Development and Analysis of Physics-based Models for Autonomous Underwater Vehicle Navigation and the Reconstruction of Underwater Images

Von der Fakultät für Ingenieurwissenschaften,

Abteilung Maschinenbau und Verfahrenstechnik der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades

eines

Doktors der Ingenieurwissenschaften

Dr.-Ing.

genehmigte Dissertation

von

Julian Hoth, M.Sc.

aus

Hannover

1. Gutachter: Prof. Dr.-Ing. Wojciech Kowalczyk

2. Gutachter: Prof. Dr.-Ing. habil. Mathias Paschen

Tag der mündlichen Prüfung: 02.05.2016

Zusammenfassung

Autonome Unterwasserfahrzeuge (AUVs) haben die Art wie die Meeresumwelt untersucht, überwacht und kartographiert wird verändert. Sie bieten eine breite Palette von Anwendungen in der Forschung, beim Militär und in kommerziellen Zusammenhängen. AUVs sollen nicht nur eine bestimmte Aufgabe erfüllen, sondern sich auch an Veränderungen in der Umgebung anpassen. Typische Einflüsse sind plötzliche Seitenströmungen, Fallströme und andere Effekte, welche extrem unberechenbar sind.

Simultane Lokalisierung und Kartenerstellung (SLAM) ist ein bekanntes und gut verstandenes Problem in der Robotik. Für landgestützte Roboter in 2D-Umgebungen wird dieses Problem im Allgemeinen als gelöst angesehen. SLAM-Algorithmen für diese neigen dazu sich auf die optische Erkennung in Kombination mit Koppelnavigation und Trägheitsmesseinheiten zu verlassen. Die optischen Eigenschaften des Wassers und insbesondere Meerwassers verhindern die Nutzung etablierter optischer Erkennungsalgorithmen. Bilder in hoher Qualität mit der richtigen Farbgebung erleichtern die Erkennung von Objekten unter Wasser und können die Verwendung der für landgestützte Roboter entwickelten visuelle SLAM-Algorithmen unter Wasser ermöglichen. Daher ist geeignete Bildverarbeitung vor allem im tiefen Wasser erforderlich.

In dieser Arbeit werden physikbasierte Modelle für die Navigation autonomer Unterwasserfahrzeuge entwickelt mit einem Schwerpunkt auf schnellen Forschungs-AUVs mit Reisegeschwindigkeiten im Bereich von 5 kn bis 20 kn. Das System sollte fähig sein Störungen im Wasserfluss zu erkennen und in der Lage sein eine Kamera zur Objekterkennung, Bodenuntersuchung und vor allem für Navigationszwecke zu verwenden.

Des Weiteren sollte es möglich sein, das System in bestehende autonome Unterwasserfahrzeuge zu integrieren. Daher muss das System klein und leicht sein, so dass die Nutzlast des AUV nicht wesentlich reduziert wird. Die erforderliche Rechenleistung und der Leistungsverbrauch müssen ebenfalls klein sein, so dass die Einsatzdauer des Fahrzeugs nicht stark verringert wird. Die Algorithmen sollten außerdem schnell sein, um SLAM-Anwendung zu ermöglichen.

Im ersten Teil der Arbeit wird die Anwendbarkeit verschiedener Lernverfahren zur Bestimmung der Strömungsparameter eines umgebenden Fluids mit Hilfe des Drucks auf einen AUV-Körper anhand zahlreicher numerischer Strömungssimulationen (CFD) und unter Verwendung von Druckdaten von festgelegten Punkten auf der Oberfläche des AUV getestet. Es wird gezeigt, dass eine Kombination von Support Vector Machines (SVM) eine ausgezeichnete Wahl ist, um diese Aufgabe auszuführen.

Mit den Ergebnissen aus den Simulationen wird dann die Lage der Druckmessstellen angepasst, so dass die höchsten Druckänderungen aufgrund der Fließgeschwindigkeiten erfasst werden. Dies reduziert auch die Anzahl von Messpunkten. Es wird dann gezeigt, dass auch für die verbesserte Konfiguration Support Vector Machines die beste Wahl für die gestellte Aufgabe sind. Jedoch sind in diesem Fall weniger Maschinen erforderlich.

Im zweiten Teil der Arbeit werden verschiedene Lernmethoden für die Rekonstruktion von Unterwasserbildern angewandt. Zuerst werden Labortests unter Verwendung einer speziellen Lichtquelle, welche die Lichtverhältnisse unter Wasser imitieren, durchgeführt. Es wird gezeigt, dass eine Kombination aus der k-nächste-Nachbarn-Methode und Support Vector Machines hervorragende

Ergebnisse liefert.

Basierend auf diesen Ergebnissen wird eine experimentelle Verifikation unter erschwerten Bedingungen im trüben Wasser eines Tauchbeckens durchgeführt. Es wird gezeigt, dass die k-nächste-Nachbarn-Methode sehr gute Ergebnisse für kleine Abstände zwischen dem Objekt und der Kamera und für kleine Wassertiefen im roten Kanal liefert. Für höhere Distanzen, Wassertiefen und für die anderen Farbkanäle ist eine Kombination von Support Vector Machines die beste Wahl für die Rekonstruktion der Farbe, wie sie unter weißem Licht zu sehen sind, aus den Unterwasserbildern.

Somit wird in dieser Arbeit ein neuer Ansatz zur Navigation autonomer Unterwasserfahrzeug und der Rekonstruktion von Unterwasserbildern vorgeschlagen und entwickelt.

Abstract

Autonomous underwater vehicles (AUVs) have changed the way marine environment is surveyed, monitored and mapped. They have a wide range of applications in research, military, and commercial settings. AUVs should not only perform a given task but also adapt to changes in the environment. Typical effects are sudden side currents, downdrafts, and other effects which are extremely unpredictable.

Simultaneous localisation and mapping (SLAM) is a well-known and well-understood problem in robotics. For land-based robots in 2-D environments this problem is generally considered to be solved. SLAM algorithms for these tend to rely on optical recognition in combination with dead reckoning and inertial measurement units. The optical properties of water and especially seawater prevent the use of established optical recognition algorithms. High quality images with correct colouring simplify the detection of underwater objects and may allow the use of visual SLAM algorithms developed for land-based robots underwater. Hence, appropriate image processing is required especially in deep water.

In this thesis physics-based models for autonomous underwater vehicle navigation are developed with an emphasis on fast exploratory AUVs with cruising speeds in the range of 5 kn to 20 kn. The system should be capable of detecting disturbances in the water flow and be able to use a camera for object detection, ground survey, and especially for navigational purposes.

Furthermore, it should be possible to integrate the system into existing autonomous underwater vehicles. Therefore, the system must be small and lightweight such that the payload of the AUV is not reduced significantly. The required computational power and the power consumption must also be small such that the duration of the vehicle does not decrease strongly. The algorithms should also be fast to allow SLAM application.

In the first part of the thesis the applicability of different learning methods for determining flow parameters of a surrounding fluid from pressure on an AUV body is tested based on numerous computational fluid dynamical (CFD) simulations and using pressure data from specified points on the surface of the AUV. It is shown that a combination of support vector machines (SVM) is an excellent choice to perform this task.

With the findings from the simulations the position of pressure measurement points is then adjusted such that the most significant pressure changes due to changing flow velocities can be captured. This also reduces the number of measurement points. It is then shown that also for the improved setup support vector machines are the best choices for the given task. However, fewer machines are required in this case.

In the second part of the thesis different learning methods are applied for the reconstruction of underwater images. First, laboratory tests are performed using a special light source imitating underwater lighting conditions. It is shown that a combination of the k-nearest neighbour method and support vector machines yields excellent results.

Based on these results an experimental verification is performed under severe conditions in murky water of a diving basin. It is shown that the k-nearest neighbour method gives very good results for small distances between the object and the camera and for small water depths in the red channel. For higher distances, water depths, and for the other colour channels a combination

of support vector machines is the best choice for the reconstruction of the colour as seen under white light from the underwater images.

Thus, a novel approach to autonomous underwater vehicle navigation and the reconstruction of underwater images is proposed and developed in this thesis.

Acknowledgments

Over the past four years I have received support and encouragement from many individuals.

First of all, a very special thank you goes to my supervisor Prof. Dr.-Ing. Wojciech Kowalczyk. His guidance has been invaluable and he gave me the freedom to follow my research interests to the fullest.

I would also like to thank my colleagues - past and present - from the Chair of Mechanics and Robotics for a creative, positive, and rewarding environment. It has been great working with you.

Of my colleagues I have to mention Andreas Scholz and Stefan Westermaier in particular as they spent a Sunday in cold water taking pictures for the validation of the algorithms.

Furthermore, a thank you also goes to the TauchRevierGasometer for allowing the use of their diving facilities and to Holger Dander for the organisation.

Last but not least, I would like to thank Prof. Dr. rer. nat. Johannes Gottschling from the Chair of Mathematics for Engineers for his insights into statistical learning.

Contents

List of Figures

List of Tables

Nomenclature

α	Multiple linear regression deviation
α	Thermal expansivity $[K^{-1}]$
α_i	Lagrange multiplier
$\bar{\lambda}$	Mean wavelength $[nm]$
β_i	Multiple linear regression coefficient
$\boldsymbol{\alpha}$	Deviation vector
$\boldsymbol{\beta}$	Coefficient vector
$\boldsymbol{\varphi}(\boldsymbol{x})$	Mapping from input space to embedding space
$\boldsymbol{b}$	Best linear unbiased estimator
$\boldsymbol{e}$	Remainder
$\boldsymbol{e}$	Unit vector
$\boldsymbol{g}$	Vector of body forces per unit mass $[m\,s^{-2}]$
$\boldsymbol{K}$	Kernel
$\boldsymbol{u}$	Velocity vector $[m\,s^{-1}]$,$[kn]$
$\boldsymbol{u}'$	Velocity fluctuations $[m\,s^{-1}]$,$[kn]$
$\boldsymbol{w}$	Normal vector to hyperplane
$\boldsymbol{X}$	Data matrix of the independent variables
$\boldsymbol{y}$	Vector of the dependent variable
δ	Kronecker tensor
δ_j	Local gradient
η	Learning-rate parameter
γ	Kernel parameter
$\hat{y}$	Estimate of dependent variable y
λ	Error penalty parameter
λ	Wavelength $[nm]$
λ_{max}	Wavelength corresponding to the maximum penetration depth of light $[nm]$
μ	Dynamic viscosity $[kg\,m^{-1}\,s^{-1}]$

μ_t	Eddy viscosity $[kg\,m^{-1}\,s^{-1}]$
∇	Vector differential operator
ν	Kinematic viscosity $[m^2\,s^{-1}]$
$\otimes$	Tensor product
π_i	Parent nodes of random variable x_i in Bayesian network
ρ	Density $[kg\,m^{-3}]$
σ_ρ	Turbulence Schmidt number [ ]
σ_ε	k-ε turbulence model constant [ ]
σ_k	k-ε turbulence model constant [ ]
τ	Shear stress tensor $[Pa]$
τ_w	Wall shear stress $[Pa]$
Δw_{ij}	Weight correction
ε	Dissipation of turbulent kinetic energy $[m^2\,s^{-3}]$
ε	Size of support vector regression tube
φ	Neuron activation function
ϑ	Turbulence velocity scale $[m\,s^{-1}]$,$[kn]$
ξ_i	Hyperplane slack variable
a	Sigmoid function slope parameter
b	Molality $[mol\,kg^{-1}]$
b	Offset of hyperplane
b_j	Bias of an artificial neuron j
c	Specific heat capacity $[J\,kg^{-1}\,K^{-1}]$
C_μ	k-ε turbulence model constant [ ]
$C_{\varepsilon 1}$	k-ε turbulence model constant [ ]
$C_{\varepsilon 2}$	k-ε turbulence model constant [ ]
$C_{\varepsilon 3}$	Dissipation coefficient [ ]
C_w	Log-Layer constant [ ]
d	Kernel parameter

d_{dive} Diving depth $[m]$

d_{ij} Distance between data point i and sample point j

d_j Desired output of neuron j

$d_{p,max}$ Maximum penetration depth of light $[m]$

d_p Penetration depth of light in seawater $[m]$

d_t Distance travelled by light through seawater $[m]$

E Total instantaneous error energy

E_j Instantaneous error energy

e_j Error signal of neuron j

h Specific enthalpy $[J\,kg^{-1}]$

I Observed light intensity $[cd]$

I Turbulence intensity [ ]

I_0 Original light intensity $[cd]$

k Diffusion attenuation coefficient [ ]

k Neighbourhood of sample point j

k Thermal conductivity $[W\,m^{-1}\,K^{-1}]$

k Turbulent kinetic energy $[m^2\,s^{-2}]$

l Turbulence length scale $[m]$

M Margin of hyperplane

M Molar mass $[kg\,kmol^{-1}]$

p Pressure $[Pa]$

$P_{\varepsilon b}$ Buoyancy production $[kg\,m^{-1}\,s^{-3}]$

P_{kb} Buoyancy production $[kg\,m^{-1}\,s^{-3}]$

P_k Turbulence production due to viscous forces $[kg\,m^{-1}\,s^{-3}]$

p_{rel} Relative pressure at openings $[Pa]$

q Heat flux $[kg\,s^{-3}]$

S Mean rate-of-strain tensor $[s^{-1}]$

S Salinity $[g\,kg^{-1}]$

s	Specific entropy $[J\,kg^{-1}\,K^{-1}]$
S_{ij}	Rate of deformation $[s^{-1}]$
T	Temperature $[°C]$
u	Component of velocity in x-direction $[m\,s^{-1}]$,$[kn]$
u^*	Velocity scale for k-ε model $[m\,s^{-1}]$,$[kn]$
u^+	Dimensionless shear velocity [ ]
u'	Component of velocity fluctuations in x-direction $[m\,s^{-1}]$,$[kn]$
u_τ	Shear velocity $[m\,s^{-1}]$,$[kn]$
v	Component of velocity in y-direction $[m\,s^{-1}]$,$[kn]$
v'	Component of velocity fluctuations in y-direction $[m\,s^{-1}]$,$[kn]$
v_j	Input data to an artificial neuron j
w	Component of velocity in z-direction $[m\,s^{-1}]$,$[kn]$
w	Mass fraction $[g\,kg^{-1}]$
w'	Component of velocity fluctuations in z-direction $[m\,s^{-1}]$,$[kn]$
w_{ij}	Distance weighting of data point i with respect to sample point j
w_{ij}	Weight of data from artificial neuron i when passed to neuron j
x_i	Input data from an artificial neuron i
x_i	Random variable in Bayesian network
x_j	Position of sample point j in data space
y	Surface normal coordinate $[m]$
y^*	Dimensionless surface normal coordinate for k-ε model [ ]
y^+	Dimensionless surface normal coordinate [ ]
y_j	Output data of an artificial neuron j
y_j	Output of sample point j
κ	Karman constant [ ]
t	Time $[s]$

1 Introduction

1.1 Autonomous Underwater Vehicles

1.1.1 Introduction

Autonomous underwater vehicles (AUVs) are a sub-group of *unmanned undersea vehicles* (UUVs) which have dramatically changed the way marine environment is surveyed, monitored, and mapped [125]. In essence they are a type of robot operating underwater. In contrast to *remotely operated underwater vehicles* (ROVs) which are controlled by an operator [35], usually on a surface vehicle, autonomous underwater vehicles are able to perform tasks on their own, i.e. autonomously. An unmanned undersea vehicle is defined as follows [4].

"Self-propelled submersible whose operation is either fully autonomous (preprogrammed or realtime adaptive mission control) or under minimal supervisory control and is untethered except, possibly, for data links such as a fiber-optic cable."

The definition shows that there are two types of autonomous underwater vehicles. On the one hand there are AUVs that simple perform a pre-programmed task. On the other hand there are AUVs which not only perform a given task but also adapt to influences and changes in the surroundings and show some kind of "intelligence" [68]. Currently, there is a clear tendency towards the latter [68].

1.1.2 Characteristics and Applications

Most autonomous underwater vehicles are torpedo-shaped [23, 55, 88] (see also fig. (1.1)). The main reasons are that the control and behaviour of torpedos are well known, they are easy and cheap in construction, and allow for a reasonable amount of mission load.

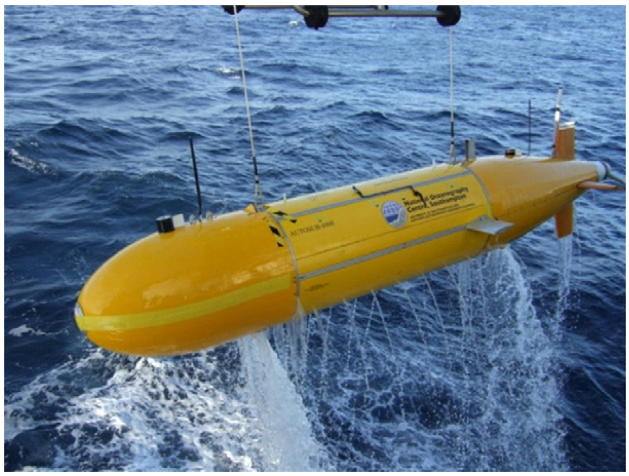


Figure 1.1: Autosub6000 lifted from the sea [88].

However, also more complex shapes are in use. AUVs like the Autonomous Benthic Explorer (ABE) (fig. (1.2)) or Sentry from the Woods Hole Oceanographic Institution (WHOI) perform exceptionally well at low speeds and in complex terrain due to their special design [128, 129]. Gliders usually have large wings [47, 70, 123]. And often shapes from nature (biomimetic vehicles)

are used in research to investigate different ways of propulsion or to understand how fish and mammals move in water [114, 126, 127].

Figure 1.2: Autonomous Benthic Explorer (ABE) from WHOI [128].

Autonomous underwater vehicles have a wide range of applications in research, military, and commercial settings. They are utilised when the use of manned undersea vehicles is too dangerous, impossible, or too expensive. Typical scientific uses with a main focus on marine geoscience are [125]:

- Collection of bathymetric data
- Investigation of hydrothermal vents and submarine volcanism
- Exploration under polar ice, e.g. [74]
- Monitoring of underwater ecosystems
- Underwater archaeology, especially deep-sea underwater archaeology (for a discussion on the problems involved see [7])

Also the military shows significant interest in autonomous underwater vehicles [18, 23, 24]. *The Navy Unmanned Undersea Vehicle (UUV) Master Plan* [4] lists 9 areas of naval applications:

- Intelligence, surveillance, and reconnaissance
- Mine countermeasures
- Anti-submarine warfare
- Inspection / identification
- Oceanography
- Communication / navigation network nodes
- Payload delivery
- Information operations
- Time critical strike

Commercial applications are also slowly emerging. The oil and gas industry shows great interest in using autonomous underwater vehicles for finding new underwater oil fields and for pipeline inspection [31]. AUVs are also applied in search and rescue operations. For instance three REMUS 6000 AUVs where successfully employed in the search for the black box of the Air France Flight 447 in 2011 [67]. And a Bluefin-21 AUV was used in the search for the wreckage of the missing Malaysia Airlines Flight 370 in 2014 [98].

1.1.3 Historical Abstract

The first underwater vehicle that falls under the category of autonomous underwater vehicle was the Self-Propelled Underwater Research Vehicle (SPURV) constructed by the Applied Physics Laboratory (APL) of the University of Washington [21]. Development began in 1959 and was completed in 1963. SPURV was already able to dive to a depth of $3659\,m$ and its purpose was to investigate the physical and chemical properties of seawater. Current AUVs can dive up to depths of $6000\,m$ and their diving time can reach several months [23, 88].

But it took until 1979 for research and development of AUVs to become more prominent. The 1980's saw more that 15 new AUVs with various academic, industrial, and military applications. A concise list of early AUVs can be found in [21].

About 70 % of early AUV development was directly or indirectly funded by the military. However, this has changed and commercial and research applications play a much more significant role.

1.1.4 Current Research

Currently, there are three main research focuses with respect to autonomous underwater vehicles. These are:

- Simultaneous localisation and mapping (SLAM)
- Autonomous decision making
- Miniaturisation

Simultaneous localisation and mapping[1] is a well-known and well-understood problem in robotics. For land-based robots in 2-D environments this problem is generally considered to be solved. Also land-based navigation in a 3-D environment is possible [94]. However, the underwater 3-D environment poses serveral difficulties [18].

Land-based SLAM algorithms tend to rely on optical recognition in combination with dead reckoning and inertial measurement units [82, 87]. The optical properties of water and especially seawater [64] prevent the use of established optical recognition algorithms.[2]

Furthermore, dead reckoning is heavily affected by influences of the surrounding environment. While for land-based robots these influences are usually well-known and can be dealt with to some extend (although not completely), sudden side currents, downdrafts, and other effects are

[1] More on SLAM can be found in section 1.3.
[2] Details on underwater image processing are discussed in section 1.4.

extremely unpredictable underwater and make dead reckoning even more unreliable than it is on land.

Inertial measurement units work well and are employed successfully in submarines [2]. Unfortunately, accurate inertial measurement units tend to be quite large, heavy, and expensive and are therefore not the first choice for AUVs. Units that are small enough for AUVs are not as accurate. This poses a problem especially due to the fact that there is an interest in small-sized AUVs.[3]

Autonomous decision making is not limited to underwater robots but of great interest in robotics in general. But due to the fact that the underwater environment is not as well-known and understood as the environment on land this is much harder to implement [108]. A focus is on establishing decision systems for cooperative behaviour of groups of robots [101].

Miniaturisation is an issue as most AUVs are quite large (about $5\,m$ length) and heavy. However, large vehicles are less manoeuvrable than small vehicles. In addition smaller AUVs may enter confined spaces like a shipwreck or an underwater cave. Difficulties arise as energy storage devices take up a significant proportion in a typical AUV [54]. In order for miniaturisation to be feasible the energy consumption of the components, notably the motors, needs to be reduced. In addition the energy density of the storage devices may be increased.

1.1.5 Future Developments

The demand for autonomous underwater vehicles is very likely to increase in the future, especially since rising fuel prices make AUVs more cost effective than traditional surveys [124].

Furthermore, there is a demand in industry for AUVs that can dive to much higher depths to reach oil and gas deposits in these regions [80]. The hybrid AUV / ROV Nereus from the Woods Hole Oceanographic Institution with a depth rating of $11000\,m$ was a first step towards fulfilling this demand. Unfortunately, it could not withstand the high pressure at $10\,km$ depth and imploded in May 2014 near New Zealand.

Another trend in AUV development is to move toward higher resolution mapping and imaging data [125]. This requires more accurate sensor systems and algorithms to deal with underwater images.

1.2 Underwater Navigation

1.2.1 Introduction

GPS localisation is not possible for undersea vehicles as radio waves do not travel through sea water very far due to high absorption. A wide variety of other systems are available for underwater navigation. Fig. 1.3 shows an outline of various navigation methods for autonomous underwater vehicles.

Usually, a set of different systems is used in one AUV and combined by means of sensor fusion algorithms [14]. In the following the methods that can be used by a single independent AUV are briefly described. These are: inertial / dead reckoning, acoustic, and magnetic methods.

[3] More on underwater navigation in section 1.2.

Underwater image processing is described in more detail in section 1.4. Transponder and modem methods require additional systems and are therefore not detailed here[4].

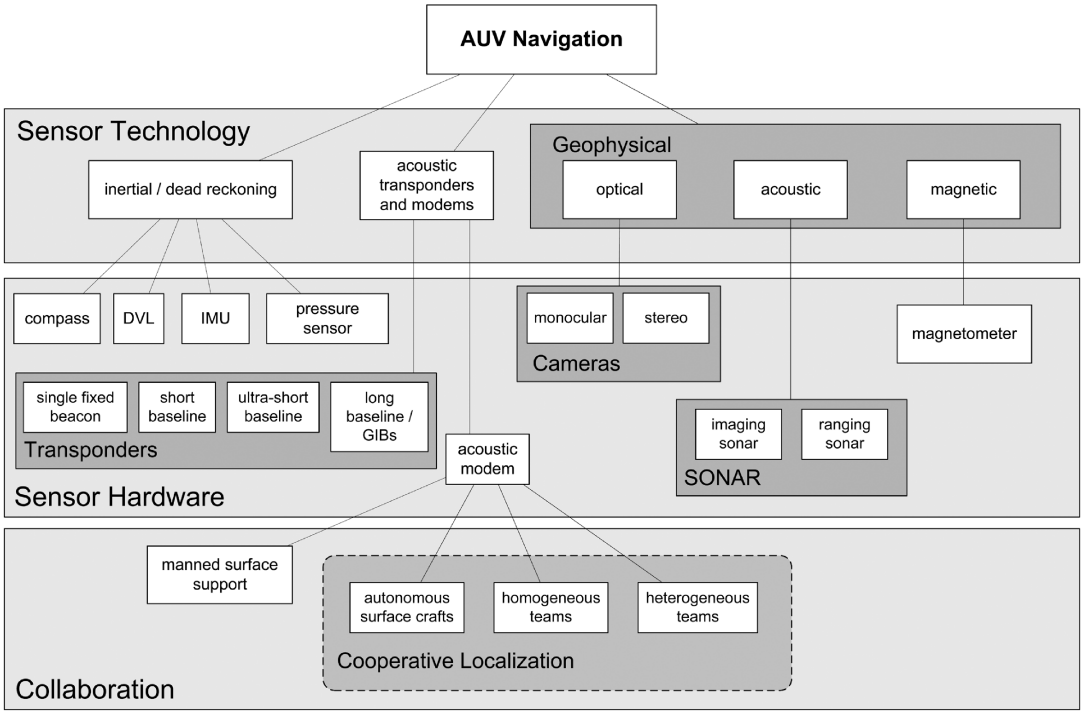


Figure 1.3: Outline of underwater navigation methods [97].

1.2.2 Dead Reckoning With a Compass

Dead reckoning uses a gyrocompass and the measured speed of the autonomous underwater vehicle to estimate its current position. Basically, a simple linear integration of the velocity over time from some previous (known) position to the current position is calculated. This method is quite simple and easy to implement. However, there are a number of sources of errors associated with it such that the resulting position may be very inaccurate [1].

Two sources of error are related to the magnetic field influencing the gyrocompass: magnetic declination and magnetic deviation. Magnetic declination is a property of the Earth's magnetic field. It describes how the measured direction toward north measured by the gyrocompass deviates from the actual direction toward the North Pole depending on the position of the vehicle. Magnetic deviation is a property of the vehicle's magnetic field due to metal components, electronic circuitry, etc. As the magnetic field of the vehicle interacts with the Earth's magnetic field the deviation is not constant but a function of the orientation of the vehicle [3].

Another source of error is the determination of the speed of the vehicle. The speed obtained with a pitometer log is the speed through water which usually deviates significantly from the speed over ground (the actual speed of the vehicle relative to the ocean floor). To obtain a more accurate result the set and drift (current) has to be take into account [1].

In conclusion it can be stated that dead reckoning works well if (and only if) some additional knowledge (magnetic field, current) exists. With surface ships and submarines this may be possible to some extent. However, with unmanned vehicles this knowledge is usually not available.

[4] An extensive review on AUV navigation and localisation can be found in [97].

1.2.3 Magnetometer

Small magnetometers can be found in many current mobile phones. The motion of the sensor through the earth's magnetic field is measured for instance with a fluxgate sensor [37]. For navigational purposes one needs to cope with the highly variable nature of the magnetic field and the influence of the AUV as was stated before with regard to the compass [38]. By building an a priori magnetic field map it should be possible to locate the AUV [97]. However, this is only a proposition at the moment and subject of ongoing research [53]. Currently, magnetometers are mostly used to map the magnetic signatures of ships.

1.2.4 Pressure Sensor

A pressure sensor can be used to measure the current depth of an autonomous underwater vehicle. However, it cannot be used on its own for navigation. Instead it supplements other methods. As the pressure gradient in water is quite high an accuracy of about $0.1\,m$ is possible [97].

1.2.5 Inertial Navigation System

The position obtained from an inertial navigation system is significantly more accurate than the one from dead reckoning. Inertial navigation uses the acceleration and the angular velocity of a vehicle to obtain the position via integration over time (for the position a double integration is required). The system consists of one or more inertial measurement units (IMU) and may be supplemented by other systems, for instance a Doppler velocity log (DVL), to increase accuracy.

Inertial measurement unit is a term used to describe a set of six sensors (three accelerometers and three gyroscopes) arranged such that the accelerometers and the gyroscopes are orthogonal to each other respectively. This way information is obtained for all six degrees of freedom [75].

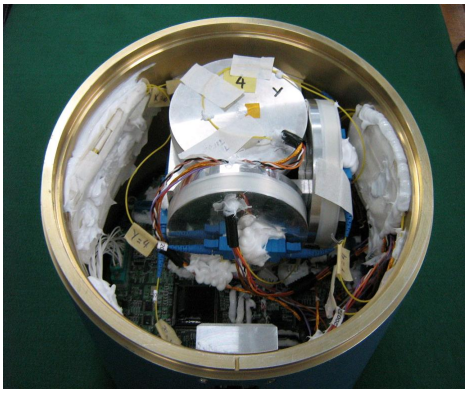


Figure 1.4: Optical MAR-1 IMU [Brazilian Air Force].

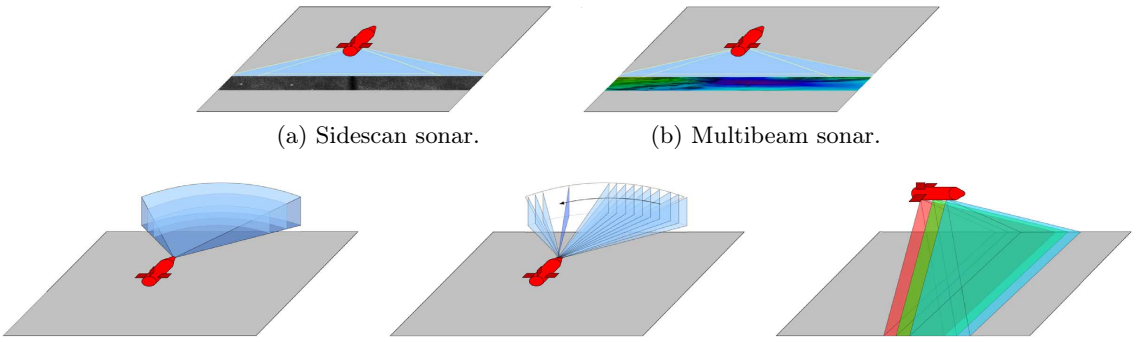
Nowadays, small-sized inertial measurement units are cheap and part of most mobile phones. These units have dimensions of just a few millimetres and require very little energy. However, these units are not accurate enough for underwater navigation (sensor drift up to $1\,^\circ/s$).

Inertial measurement units for navigational purposes are significantly larger (dimensions about $20\,cm$) than chip-based IMUs. High-end IMUs with fibre optic gyroscope (e.g. MAR-1 guidance unit, fig. 1.4) have a sensor drift of about $10^{-3}\,^\circ/h$ (recently developed instruments show a sensor

drift that is even as low as $10^{-4}\,°/h$). This translates into a positional error of less than $100\,m$ per hour [118].

1.2.6 Sonar

A sonar detects and locates objects in water using sound waves. There are a number of different sonar systems available: sidescan sonar, multibeam sonar, forward looking sonar, mechanical scanning and imaging sonar, and synthetic aperture sonar [97].



(a) Sidescan sonar. (b) Multibeam sonar.

(c) Forward looking sonar. (d) Mechanical scanning sonar. (e) Synthetic aperture sonar.

Figure 1.5: Types of sonars [97].

A sidescan sonar (fig. 1.5(a)) produces a sound beam which is directed perpendicular to the direction of travel of the AUV. The beam is reflected at the seafloor and a cross-track slice of the structure of the sea bed is obtained [103]. Consecutive slices are assembled to produce a 2-D map [32]. Sidescan sonars work at AUV speeds up to $10\,kn$ and the resolution of the resulting map is inversely proportional to the range of the sound beam [97].

With a multibeam (fig. 1.5(b)) sonar several sound beams are produced simultaneously. The data collection is therefore more efficient [97]. Also a multibeam sonar produces an elevation map (so-called 2.5-D bathymetric map) [97]. However, the data produced by sidescan sonars is of better quality as the alongtrack beamwidth is lower than for multibeam sonars [99].

The purpose of forward looking sonars is very different to sidescan or multibeam sonars. As shown in fig. 1.5(c) the sound beams are directed toward the horizontal. Therefore, forward looking sonars are used for mapping vertical features [97]. A typical application is the inspection of man-made structures, for instance ship hulls [117]. Mechanical scanning sonars are very much like forward looking sonars. But instead of multiple beams projected forward there is only a single beam which is continuously panning from port to starboard and back (see fig. 1.5(d)).

Instead of using a large array of emitters only a small one is required for synthetic aperture sonars. By taking the motion of the AUV into account and careful processing of the data a large virtual array is created (fig. 1.5(e)). To achieve this considerable computational power and very precise vehicle motion is required [97]. The latter is difficult to achieve in the underwater environment due to the dynamic nature of the surroundings.

1.2.7 Doppler Velocity Log

As mentioned above inertial navigation systems may be supplemented by a Doppler velocity log (DVL). This is a specialised application of an acoustic Doppler current profiler (ADCP). The ADCP is a type of active ultrasonic sonar which measures the velocity of water up to 1000 m in front of the ADCP head (fig. 1.6). The system makes use of the reflection of sound waves at suspended particles in the water in combination with the Doppler effect [93].

Figure 1.6: 4 beam ADCP head [NOAA].

If the sound waves reflected from the sea floor are analysed instead of those reflected from the suspended particles in the water the ADCP becomes a DVL and the velocity of the system itself, i.e. the AUV, can be determined. Combined with an inertial measurement unit the position of the vehicle can be obtained.

1.3 Simultaneous Localisation and Mapping

1.3.1 Introduction

Simultaneous localisation and mapping (SLAM) is a method for robots and specifically for autonomous robots to *create a map of an unknown environment* and to find the *current position (or pose) of the robot itself within the environment* at the same time, usually beginning without any knowledge on the map and the location [48].

The first formulations of the problem with solutions based on extended Kalman filters where done in the early 1990s [81, 110]. This was follow by a number of different SLAM algorithms as interest increased greatly in the 2000s [85]. As mentioned before, SLAM is now considered to be solved for 2-D environments in general with land-based navigation in a 3-D environment also being possible [94]. An excellent review of land-based SLAM algorithms can be found in [42].

The most important characteristic of SLAM algorithms that has to be understood is that there exists no single SLAM merhod that can be applied to all situations. Every SLAM algorithm is unique to the environment, the sensors of the robot, and the operation conditions [48]. Therefore, the existing algorithms that work very well for land-based robots cannot simply be translated directly for use underwater. All three factors (environment, sensors, operational condition) are, obviously, significantly different.

1.3.2 SLAM in the Underwater Environment

The underwater environment is particularly challenging for autonomous navigation [68]:

- Electromagnetic signals cannot penetrate seawater very far (no GPS signal)
- Limited bandwidth, scale, and reliability of acoustic communication [29]
- Observationally limited environment, e.g. low visibility
- Constantly changing environment due to sediment deposits, tidal forces, etc.

As sonar technology is well understood and used on submarines a logical step is to apply this knowledge to autonomous underwater vehicles as well. Up to the mid-2000s the focus was on the use of acoustic data for SLAM [84, 115, 121].

However, data obtained from acoustic ranging has low spacial resolution, obscured object boundaries, and strong disturbances. The two main reasons for this are the constantly changing surroundings (particularly the seafloor) and the ability of sound waves to penetrate certain materials such that the reflections are not necessarily at the actual boundaries of objects. The resulting data is therefore difficult to interpret and often inaccurate [130]. A more resent paper uses extended Kalman filters and an improved ant colony algorithm to tackle these problems [130]. However, the approach was only tested in the very controlled environment of a water tank where these difficulties usually do not appear.

It is for the above reasons that in the late 2000s several research groups started investigating the possibility of using visual SLAM for underwater vehicles. There are several advantages of vision-based systems for underwater SLAM [104]:

- Cameras are widely available and significantly cheaper that standard sonar equipment
- Video images are preferred for scientific exploration
- High-resolution images can be obtained with current camera systems
- Many visual SLAM algorithms should (with some modifications) be applicable to underwater images

However, there are some properties of underwater images that make a direct translation of visual SLAM algorithms difficult [104]:

- Sparse features
- High level of noise and distortions
- Uniform colouring of the environment

It is for the last two reasons that most groups working on underwater SLAM focus on areas with little water depth (a few meters) and relatively clear water. Noise is a typical problem in murky water and uniform colouring occurs especially in deep sea[5]. This way only the sparse features have to be dealt with.

Significant progress has been made in using stereoscopic vision for underwater SLAM in surveys [85, 83]. In a survey the AUV covers a specific area and usually returns to some previous positions

[5]Tackling the problems of murky water and colour is discussed in section 1.4

several times. This way, feature matching can be used to correct the supposed path of the vehicle (for instance contributions have been made in mapping a benthic habitat at Ningaloo Reef [122]). Unfortunately, this does not help with extended missions where the AUV may not return to a previous position at any time. Also the corrections cannot be made continuously but only when matching features are found.

1.4 Underwater Image Processing

1.4.1 Introduction

Objects look very different in the underwater environment compared to their appearance in sunlight. The main reason is that the penetration of light through seawater is highly dependent on the wavelength (energy) of the light [64]. Suspended particles in the water can further decrease the overall quality of underwater images [17, 30].

High quality images with correct colouring simplify the detection of underwater objects and may allow the use of visual SLAM algorithms developed for land-based robots underwater. Hence, image processing is required to obtain images of high quality and correct colouring. Over the last decade significant progress has been made in this direction. Unfortunately, the existing algorithms still have limitations and can only be used under specific circumstances (see below).

1.4.2 Optical Properties of Seawater

Optical properties of seawater can be divided into two classes [92]. Inherent optical properties (IOPs) are only depending on the material, i.e. seawater, while apparent optical properties (AOPs) additionally depend on the geometric structure of the light field. IOPs are absorption coefficient, volume scattering function, index of refraction, beam attenuation coefficient, and single-scattering albedo. AOPs include irradiance reflectance, average cosines, and diffuse attenuation coefficients [92].

Seawater is a mixture containing various dissolved and suspended substances (mostly salts) besides pure water[6] [64]. The exact composition varies in time and depending on the location, although the chemical composition of ocean water is in general more uniform on large scales than the chemical composition of coastal waters which strongly depends on local factors. Due to this varying composition also the optical properties of seawater are different depending on time and place [92].

In the following the physical definitions and meanings of the properties mentioned above are not given[7]. The implications for underwater imagery, namely colour and range, are discussed instead.

The main factor of interest for the influence on colour is the penetration depth d_p which is a strong function of the wavelength λ. This can clearly be seen in fig. 1.7.

For typical coastal waters the maximum penetration depth $d_{p,max,coastal}$ is reach for a wavelength of $\lambda_{max,coastal} \approx 560\,nm$ with a low penetration depth in the purple range. This results in a

[6] For the chemical composition of ocean water see section 2.2.

[7] For a detailed description of optical properties of seawater [92] is recommended.

slightly dark greenish colour of the water. The colour of objects underwater is altered accordingly. This can, for instance, be observed for the Baltic Sea.

For the open ocean the maximum penetration depth $d_{p,max,ocean}$ occurs at a wavelength of $\lambda_{max,ocean} \approx 480\,nm$ (see also [111, 92]). A low penetration depth in the red range can also be observed and the penetration depth in the purple range is much higher than for coastal waters. This results in a dark bluish colour of the water and underwater objects which is typical, for instance, for the Atlantic Ocean.

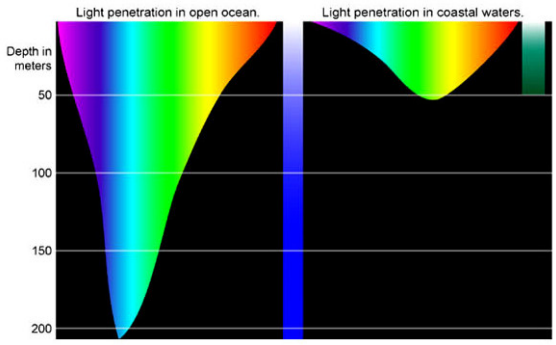


Figure 1.7: Light penetration in ocean water and coastal waters [27].

It can also be observed that the maximum pentration depth for the open ocean is about four times larger ($d_{p,max,ocean} \approx 220\,m$) compared to coastal waters ($d_{p,max,coastal} \approx 60\,m$). The main reason are suspended particles which significantly reduce visibility [92].

1.4.3 Underwater Image Enhancement

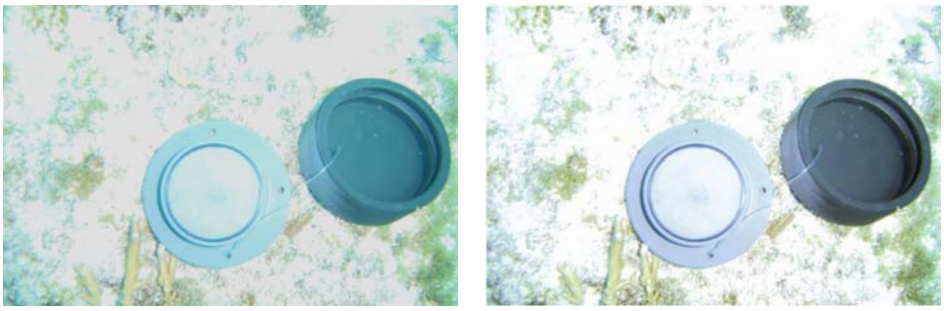
In her PhD thesis [64] Åhlén develops a number of algorithms to improve the colouring of underwater images. One algorithm uses the absorbance coefficient k for the three channels (red, green, blue) of a camera to estimate the loss of light which is then added onto the image [61]. The estimation is done using *Beer's law*

$$I = I_0 e^{-kd_t} \tag{1.1}$$

with I the observed light intensity, I_0 the original light intensity, and d_t the distance travelled by light though seawater. For a calibration object both I and I_0 are known in addition to d_t which results from the diving depth. Hence, k can be calculated and I_0, i.e. the colour corrected image, can be determined for other objects. Fig. 1.8 shows the algorithm in practise for a simple object at a depth of $12\,m$. The colour corrected image shows the object as it would be seen at a depth of $3\,m$.

A greenish tint can be observed in the original image which is typical for coastal waters as stated before. It is also of interest to note that for a depth of $12\,m$ only a small amount of the overall light spectrum is filtered by the seawater (fig. 1.7). So the "difference" between the original and the corrected image is not very large.

Åhlén also takes into account the influence of bottom refection [62]. This is dealt with by introducing a correction factor in Beer's law that is a function of the reflectance in the image

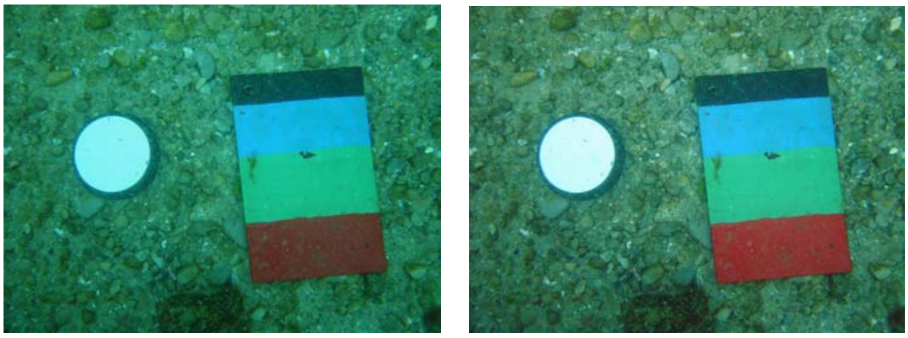


(a) Original image at 12 m depth. (b) Colour corrected image.

Figure 1.8: Colour correction using absorbance coefficients [64].

determined using a calibration object. Futhermore, some work is done on the influence of dissolved particles on the absorbance coefficient k [63].

Later in her work Åhlén realises that an analysis on the three camera channels may not be sufficient. The algorithm is expanded to include the analysis of spectral data [65, 66]. This improves the results further (fig. 1.9). However, the overall method based on Beer's law with correction factors remains the same.



(a) Original image at 6 m depth. (b) Colour corrected image (equivalent to 1.8 m depth).

Figure 1.9: Colour correction due to effects of suspended particles [64].

A very different approach to underwater image enhancement is taken by Bazeille et al. [17]. Their proposed pre-processing filter (this indicates that the filter is only meant as a first step) requires no adjustable parameters and is therefore independent of the actual lighting conditions on site. Only the information provided by the image itself is used. The filter consists of a number of sub-filters[8] and concentrates on improving the overall image quality removing blur, haziness, etc. as follows.

- Removing potential moiré effect
- Resizing and extending the image symmetrically to obtain a square image (for FFT and FWT)
- Conversion from RGB to YCbCr
- Homomorphic filtering

[8] A comparison of three of the sub-filters can be found in [96].

- Wavelet denoising
- Anisotropic filtering
- Adjustment of image intensity
- Conversion from YCbCr to RGB
- Equalisation of colour mean

Colour correction is not done extensively, except for the equalisation of the colour mean as the last step, although the resulting images (fig. 1.10) are also improved in colour to a great extent.

Figure 1.10: Pairs of images before (left) and after correction (right) [17].

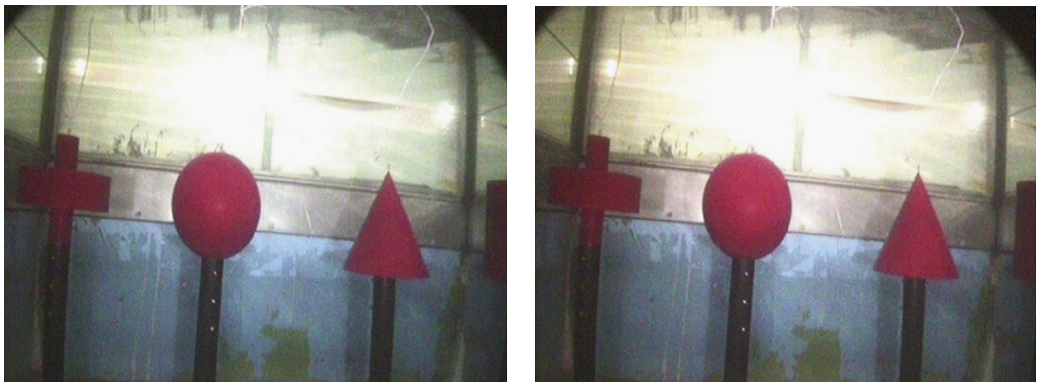
As with [64] the images used have a greenish tint indicating coastal waters. Although it is not mentioned, again diving depth may be assumed (as divers appear in one of the images). However, the difference between the original and the filtered images is far more extensive than for [64] making this pre-processing filter superior.

Iqbal et al. [69] again focus more on the colouring of underwater images. Their approach consists of three steps, namely contrast stretching of the RGB image and saturation and intensity stretching of the image converted to HSI. The method is simpler than the one by Åhlén [63]. However, the same limitations apply as stretching requires a certain minimum amount of intensity information retained in the image. This is only true for small depths and the examples given in the paper fulfil this requirement.

More recent approaches of underwater image processing include the use of geometric parameters in combination with the Jaffe-McGlamery image model [79], empirical mode decomposition [30], and region-specialised restoration [33].

The *Jaffe-McGlamery image model* is essentially an extension of Beer's law where the irradiance is decomposed into a direct component, a forward-scatter component, and a backscatter component [72]. The three components are then handled separately for the underwater image enhancement. In [79] this is combined with parameters describing the geometric relations between the camera, the light source, and the object. As one may observe (fig. 1.11) the difference between the original image and the processed image is almost insignificant. Lee et al. do not show whether

their proposed approach is also applicable for more extreme cases as used by Åhlén [64] or Bazeille et al. [17]. An additional disadvantage is the fact that a number of geometric measurements have to be taken in order for the method to be applicable.



(a) Original image. (b) Colour corrected image.

Figure 1.11: Colour correction using geometric parameters and Jaffe-McGlamery image model [79].

Celebi and Ertürk [30] use the same original images as Bazeille et al. Hence, a direct comparison of their methods is possible. The algorithm features the use of empirical mode decomposition (EMD). The data of the three colour channels of the image is decomposed into intrinsic mode functions (IMFs) and a residual. The decomposition is done repetitive in such a way that the residual does not contain any extrema. The enhanced image is then a weighted sum of the IMFs. The required weights are determined automatically using a generic algorithm.

Figure 1.12: Enhanced underwater images using empirical mode decomposition [30].

Fig. 1.12 shows the resulting images. When compared to the results by Bazeille et al. (fig. 1.10) one may observe that the colour reconstruction is, in general, better for the new algorithm by Celebi and Ertürk. On the other hand, the overall image quality (reduction of fussiness, haziness, etc.) is better for the algorithm by Bazeille et al. However, one must keep in mind that colour reconstruction is not the primary goal for Bazeille et al. So both algorithms complete their respective tasks very well.

Chen et al. focus on region-specialised restoration [33]. The argument is that different parts of an image require different treatments due to strong variations in optical properties of objects in the image[9]. Although the argument is valid and the proposed algorithm shows good results (it is also based on an exponential attenuation law), their approach has one vital flaw: part of the background has to be selected manually as a reference sample. The method is therefore not applicable to autonomous underwater vehicles.

[9] In section 3.2 this will also be discussed.

1.5 Objective

1.5.1 Underwater Navigation

In order to work autonomously for an extensive period of time in an unknown underwater environment autonomous underwater vehicles need to perform simultaneous localisation and mapping and react to changes in the surroundings. To achieve this the positional error of the AUV must be as small as possible and the influences of the environment on the AUV must be detected accurately. This requires a very precise real-time capable navigation system or combination of navigation systems.

In the underwater surroundings typical influences are sudden side currents, downdrafts, and other effects which are extremely unpredictable. In order to navigate properly, these effects need to be detected. However, the disturbances in the water flow cannot be measured using existing systems. Only the indirect effects, e.g. a change in acceleration, can be observed.

It is of interest to detect the disturbances directly through measurements to increase the accuracy and reliability of the navigation system of an AUV with an emphasis on fast exploratory AUVs with cruising speeds in the range of $5\,kn$ to $20\,kn$. Hence, a system is required which is capable of this task. Furthermore, it should be possible to integrate the system into existing autonomous underwater vehicles. Therefore, the system must be small and lightweight such that the payload of the AUV is not reduced significantly. The required computational power and the power consumption must also be small such that the duration of the vehicle does not decrease strongly. Furthermore, the system should also be accurate for fast autonomous underwater vehicles.

As the disturbances yield an instantaneous change of the pressure that is exerted on the surface of the AUV body it is proposed to place sensors at specific points on the AUV body to measure the pressure at these positions. The aim of this thesis is to investigate how machine learning can be used to obtain flow parameters of the surrounding fluid from pressure data and which machine learning algorithms are appropriate for this task. Also the influence of data pre-processing on the performance of the learning machines is to be investigated. As the relation between the pressure on the AUV body and the fluid flow is quite complex a model-based approach is not sufficient.

Determining flow parameters from pressure is the reverse of the standard method, where the flow parameters are known and the pressure distribution is calculated. To this end an extensive number of different physics-based flow situations are simulated using computational fluid dynamics, the pressure distribution and the flow parameters are obtained, and analysed using different learning algorithms.

1.5.2 Image Processing

Objects look very different in the underwater environment compared to their appearance in sunlight. High quality images with correct colouring simplify the detection of underwater objects and may allow the use of visual SLAM algorithms developed for land-based robots underwater. This will increase the capabilities of autonomous underwater vehicles signifinantly as the algorithms for land-based robots are already well-established and advanced. Hence, image processing is required to obtain images of high quality and correct colouring.

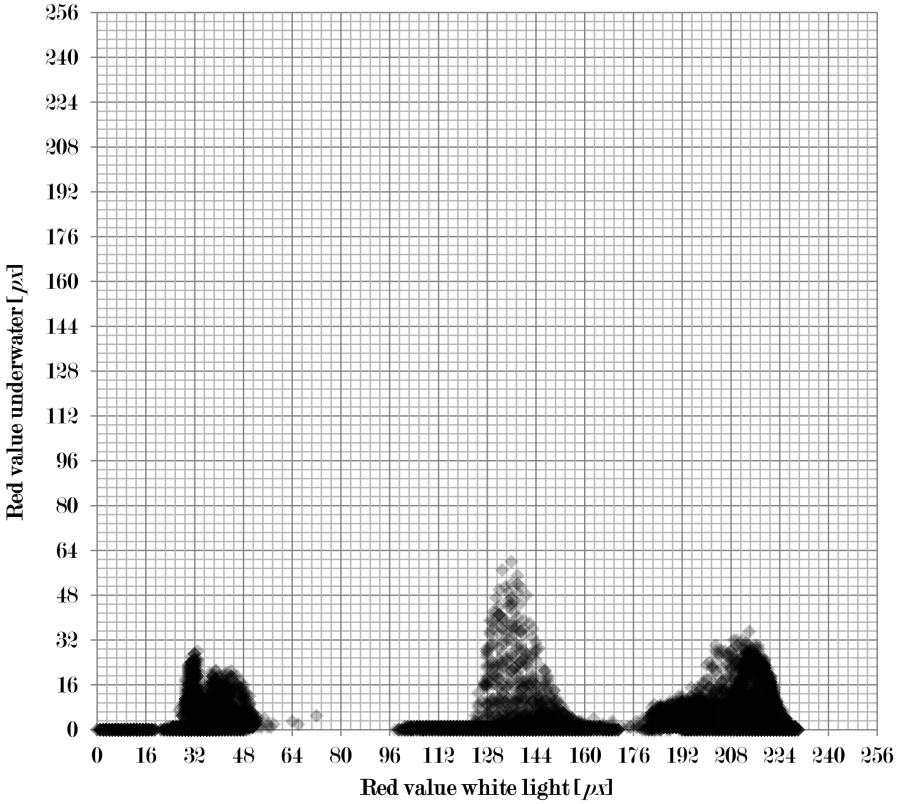


Figure 1.13: Comparing colour values of the red channel.

At great depths and for objects which are sufficiently far away from the camera the filtering due to the optical properties of ocean water is significant such that different colours are strongly tainted and can no longer be distinguished. Fig.'s 1.13, 1.14, and 1.15 show the relation of the colours of various objects as seen underwater (on the ordinate) and under sunlight (on the abscissa). It is clear from the graphs that the relation is very complex and model-based approaches are not sufficient to describe this relation.

Hence, an algorithm is needed which is capable of calculating the colours as seen under sunlight from the colour in the underwater scenery. The algorithm's required computational power must be small such that the duration of the AUV is not reduced due to a high-end processor. The algorithm should also be fast and real-time capable for the SLAM application.

The aim of this thesis is to investigate how machine learning can be used to transform image data obtained under underwater lighting conditions and which machine learning algorithms are appropriate for this task. In addition the influence of data pre-processing on the performance of the learning machines is to be investigated.

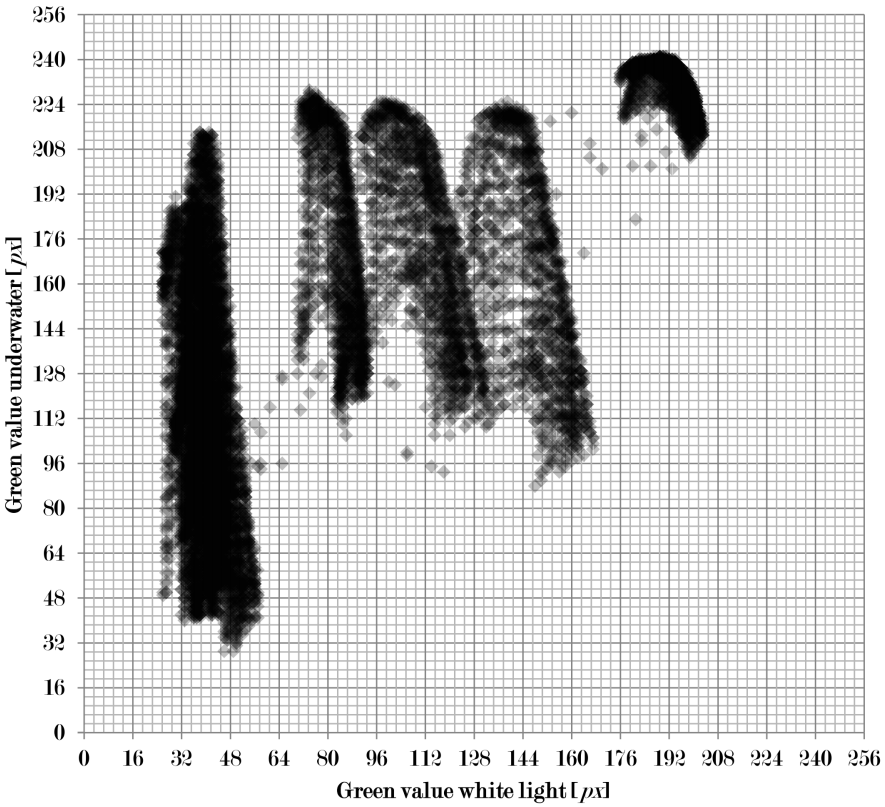


Figure 1.14: Comparing colour values of the green channel.

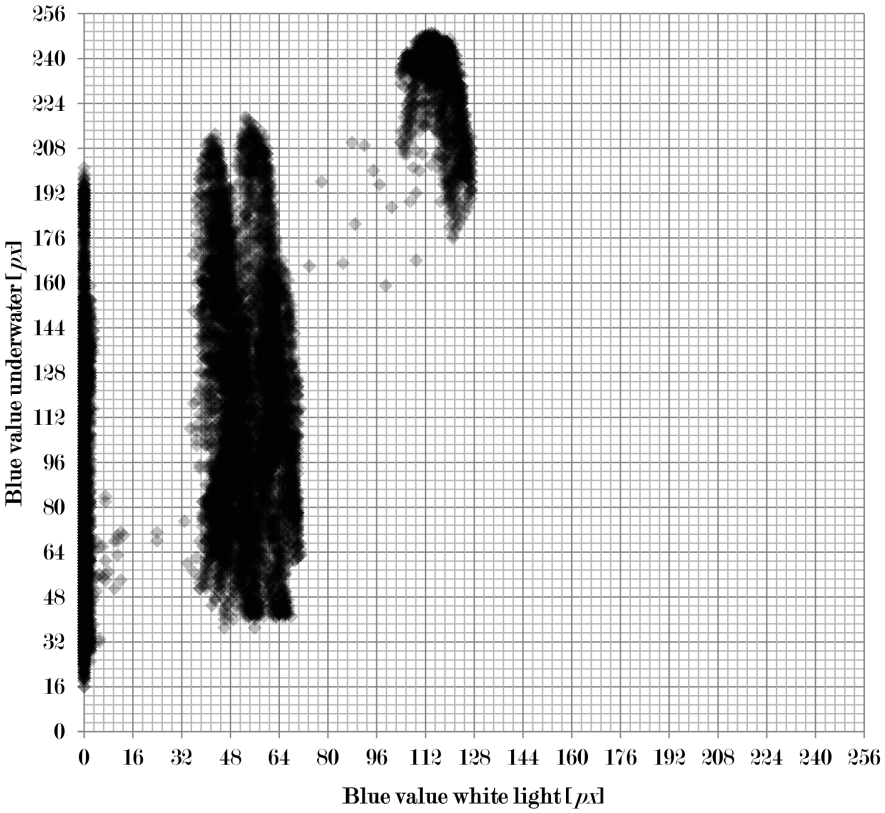


Figure 1.15: Comparing colour values of the blue channel.

2 Materials and Methods

2.1 Machine Learning

2.1.1 Introduction

In the following five machine learning methods, which are used in this study, are introduced. The distinction between pure statistical methods and pure learning methods is not always clear. Hence, also the term *statistical learning* is often used for some methods (see also [57]). However, machine learning algorithms tend to have some empirical or heuristical, as well as some discrete or combinatorial characteristics [102].

2.1.2 Artificial Neural Networks

Artificial neural networks (ANN) are information processing algorithms that are modelled after the way brains work [9]. The first formulations of this method were already made in 1943 by W. McCulloch and W. Pits [86]. The idea was to build a computational model of the nervous system in the brain [8]. With this method it is possible to obtain patterns and information from complex data.

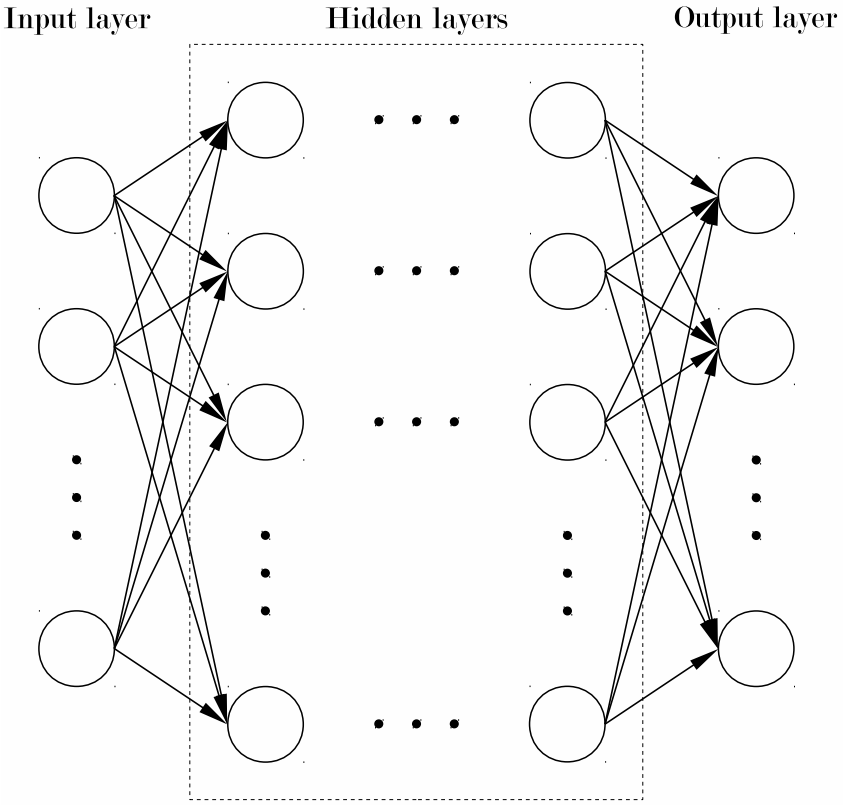


Figure 2.1: Artificial neural network.

Neural networks consist of strongly interconnected nodes called artificial neurons (see fig. 2.2). These neurons are organised into layers: one input layer, one output layer, and one or more hidden layers in-between (see fig. 2.1). The number of nodes in the input and output layer is given due to the nature of the data being processed. The number of nodes in the hidden layers as well as the number of hidden layers is variable [6]. This depends of the type and complexity

of the model to be built.

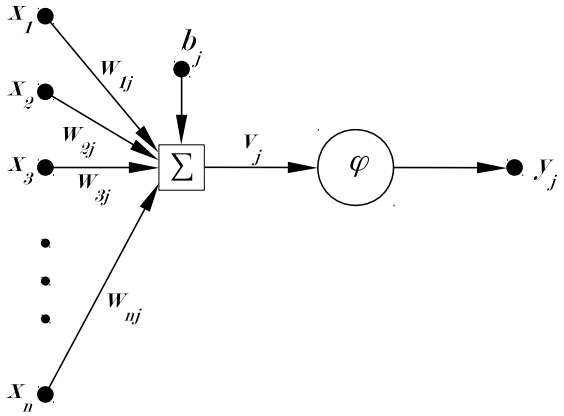


Figure 2.2: Artificial neuron.

In an artificial network data x_i is passed to a neuron j from nodes i in the previous layer. All data x_i is weighted by w_{ij}. w_{ij} may be positive or negative. In addition every node has a bias b_j which can also be positive or negative. The input v_j to node j is therefore (see fig. 2.2) [58]

$$v_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{2.1}$$

It should be noted that data is usually normalised to $[0, 1]$ or $[-1, 1]$. This is to handle significant differences in magnitude of input and output data which may occur due to a possible vastly different nature of the data to be handled by the artificial neural network.

The output y_j of node j is then some function φ of the input v_j. φ is called the activation function of the artificial neuron.

$$y_j = \varphi(v_j) \tag{2.2}$$

There are essentially two fundamental types of activation functions [58]. On the one hand threshold functions may be used where the neuron is either activated or not. This can be achieved by way of the Heaviside function for an output range $[0, 1]$

$$\varphi(v_j) = \begin{cases} 1 & \text{if } v_j \geq 0 \\ 0 & \text{if } v_j < 0 \end{cases} \tag{2.3}$$

or using the signum function for an output range $[-1, 1]$

$$\varphi(v_j) = \begin{cases} 1 & \text{if } v_j > 0 \\ 0 & \text{if } v_j = 0 \\ -1 & \text{if } v_j < 0 \end{cases} \tag{2.4}$$

On the other hand it may be desireable to have a smooth transition between the two states "node not active" and "node active". In this case sigmoid functions with slope parameter $a > 0$ are used. For an output range $[0, 1]$ the logistic distribution function

$$\varphi(v_j) = \frac{1}{1 + e^{-av_j}} \tag{2.5}$$

and for the output range $[-1, 1]$ the hyperbolic tangent

$$\varphi(v_j) = \tanh(av_j) \tag{2.6}$$

are possible candidates[10].

Sigmoid functions have a significant advantage over threshold functions as they are differentiable which is imporant for learning (see below). Hence, they are much more common as activation functions [58].

Teaching neural networks is done by forming or removing connections, changing the weights, changing the biases of neurons, and adding or removing neurons such that for a given validation set the error becomes minimal. However, in many cases the overall shape of the neural network is fixed and only the weights and biases are used for learning [6].

A common method for training is the back-propagation algorithm. Hereby, the error derivative of the weights, i.e. the change of error depending on the change of weight, is determined backwards by starting with the total error at the output layer and the moving through the network toward the input layer [116]. This is done repeatedly until either the rate of change of the errors or the error derivatives become sufficiently small.

Let the error signal e_j of a neuron j with respect to the desired output d_j be

$$e_j = d_j - y_j \tag{2.7}$$

Then the instantaneous error energy E_j for that node is

$$E_j = \frac{1}{2}e_j^2 \tag{2.8}$$

For the whole net one can define the total instantaneous error energy E as the sum of the instantaneous error energies of the neurons in the output layer O [119].

$$E = \sum_{j \in O} E_j \tag{2.9}$$

One can now write the error derivative $\frac{\partial E}{\partial w_{ij}}$ using the chain rule [58].

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial e_j}\frac{\partial e_j}{\partial y_j}\frac{\partial y_j}{\partial v_j}\frac{\partial v_j}{\partial w_{ij}} \tag{2.10}$$

With eqn. 2.2 and eqn.'s 2.7 to 2.9 in eqn. 2.10 one obtains:

$$\frac{\partial E}{\partial w_{ij}} = -e_j y_j' x_i \tag{2.11}$$

[10] Note that the Heaviside function and the signum function are also sigmoid functions for the limiting case $a \to \infty$ for the logistic distribution function and the hyperbolic tangent respectively.

The delta rule leads to the weight correction Δw_{ij} as

$$\Delta w_{ij} \;=\; -\eta \frac{\partial E}{\partial w_{ij}} \;=\; \eta \delta_j x_i \tag{2.12}$$

with η the learning-rate parameter and with δ_j the local gradient:

$$\delta_j \;=\; \begin{cases} y'_j e_j & \text{if } j \in O \\ y'_j \sum\limits_k \delta_k w_{jk} & \text{if } j \notin O \end{cases} \tag{2.13}$$

where

$$y'_j \;=\; a y_j \left(1 \,-\, y_j\right) \tag{2.14}$$

in case of a logistic activation function and

$$y'_j \;=\; a \left(1 \,-\, y_i^2\right) \tag{2.15}$$

in case of a hyperbolic tangent activation function.

2.1.3 k-Nearest Neighbour

k-nearest neighbour (KNN) is a non-parametric method for classification and regression. The output y_j (classification or property value) for a given input x_j, i.e. its position in data space, is obtained from known input-output relations where the inputs are similar and close to the sample point in question. Classification is then done by majority voting and in case of a property value the average of the outputs of the neighbouring inputs is taken [50].

The result strongly depends on the choice of the neighbourhood k, i.e. up to what distance or how many neighbours are taken into account. This is illustrated in the following example (fig. 2.3).

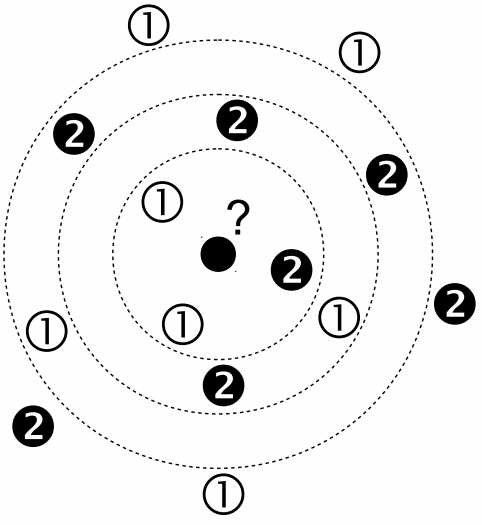


Figure 2.3: Example of k-nearest neighbour.

If the neighbourhood is taken to be $k = 3$ (inner circle) the sample point will be classified as '1' as there are more 1's than 2's. But for $k = 9$ the classification will be '2'. And if one takes $k = 6$ the classification problem will be undecidable due to the same number of 1's and 2's.

It should be noted however that in general k is uneven for classification problems [60]. This ensures that a tie in the majority voting is avoided. For regression this restriction in not necessary.

Approaching the same example as a regression problem the resulting property value of the sample point can be obtained by averaging with

$$y_j = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{2.16}$$

This yields $y_j \approx 1.3$ for $k = 3$, $y_j = 1.5$ for $k = 6$, and $y_j \approx 1.6$ for $k = 9$.

It is sensible to assume that data points which are closer to the sample point are more likely to have similar outputs [60]. Hence, the contributions of the neighbouring data points are weighted depending on the distance such that data points closer to the sample point have a stronger influence on the output. The weight w_{ij} of a data point i with respect to sample point j can be expressed as

$$w_{ij} = \frac{e^{-d_{ij}}}{\sum_{i=1}^{k} e^{-d_{ij}}} \tag{2.17}$$

where d_{ij} is the distance between the two points. The distance metric can be chosen as follows.

$$d_{ij} = \begin{cases} \sqrt{(\boldsymbol{x_j} - \boldsymbol{x_i})^2} & \text{Euclidean} \\ (\boldsymbol{x_j} - \boldsymbol{x_i})^2 & \text{Euclidean squared} \\ |\boldsymbol{x_j} - \boldsymbol{x_i}| & \text{Cityblock} \\ \max |\boldsymbol{x_j} - \boldsymbol{x_i}| & \text{Chebyshev} \end{cases} \tag{2.18}$$

of which the Euclidean distance measure is the most common [60]. It can easily be seen from eqn. 2.17 that the sum of all distance weights is

$$\sum_{i=1}^{k} w_{ij} = 1 \tag{2.19}$$

Then, the output of the sample point can be obtained from

$$y_j = \sum_{i=1}^{k} w_{ij} y_i \tag{2.20}$$

For classification eqn. 2.20 is evaluated for each class variable. The maximum result determines the classification of the sample point.

Coming back to the example (fig. 2.3) the results with (Euclidian) and without distance weighting are sumarised in tab 2.1. As one can see distance weighting has a significant influence on the classification respectively the property value of the sample point.

	Without distance weighting		With distance weighting	
	Classification	Regression	Classification	Regression
$k = 3$	$y_j = 1$	$y_j \approx 1.3$	$y_j = 1$	$y_j \approx 1.3$
$k = 6$	undecidable	$y_j = 1.5$	$y_j = 1$	$y_j \approx 1.4$
$k = 9$	$y_j = 2$	$y_j \approx 1.6$	$y_j = 1$	$y_j \approx 1.4$

Table 2.1: Example of k-nearest neighbour.

Actually finding the nearest neighbour is not an easy task. Especially in higher dimensions the search is costly from a computational point of view [95]. Therefore, a number of algorithms exist to simplify the neighbourhood search. The resulting neighbourhoods are then no longer exactly defined only by the distance metric as given in eqn. 2.18 [95].

Teaching a k-nearest neighbour system is usually done by finding the size of the neighbourhood k with the lowest error by cross-validation. Hereby, the given data is split into several sub-sets. On each of these the k-nearest neighbour algorithm is applied for a range of different values for k. The value of k achieving the lowest error is then chosen [20, 60].

2.1.4 Support Vector Machines

Support vector machines (SVM) are non-probabilistic linear classifiers. However, they can also be used for regression [43]. The fundamental idea is that data sets that belong to different classes are linearly separable by hyperplanes. If the data cannot be separated by linear hyperplanes it is to be mapped into a higher dimensional space such that it becomes linearly separable. The result is transformed back to the original input space [41].

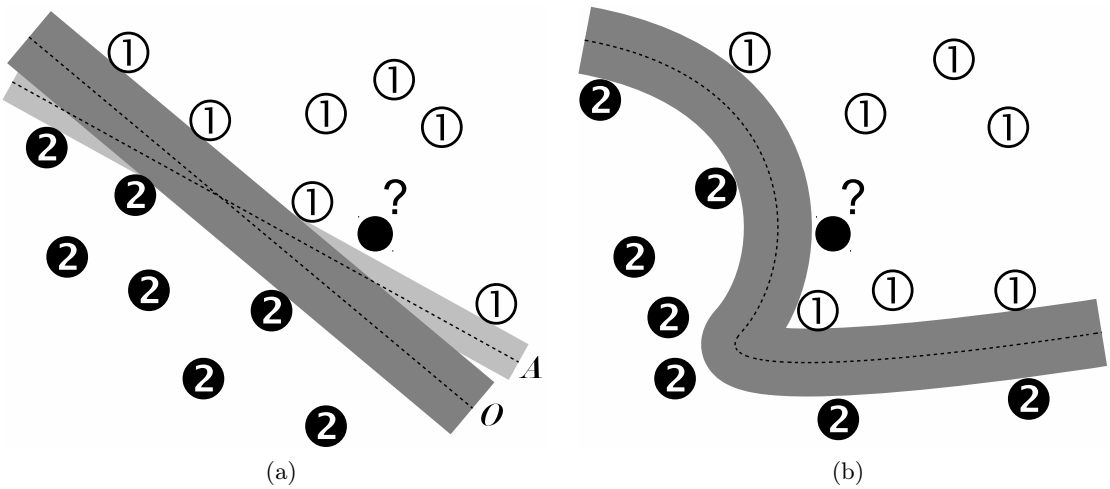


Figure 2.4: Examples of SVMs for linearly (a) and nonlinearly (b) separable data.

The discussion begins with the linearly separable case in an n-dimensional input space. Then there exists some $(n - 1)$-dimensional hyperplane of the form

$$f(\boldsymbol{x}) \coloneqq \boldsymbol{w} \cdot \boldsymbol{x} + b = 0 \tag{2.21}$$

that separates the data, e.g. plane A in fig. 2.4(a) [102]. Here $\boldsymbol{w}$ is a normal vector to the hyperplane, b is an offset, and $\boldsymbol{x}$ is an n-dimensional vector. A sample point $\boldsymbol{x}_i$ can then be

classified depending on which side of the hyperplane it is situated. For simplicity the classification is $y_i \in \{-1, 1\}$, i.e. there is a "positive" and a "negative" side of the hyperplane. One obtains

$$\begin{cases} \boldsymbol{w} \cdot \boldsymbol{x}_i + b \geq 0 & \text{for } y_i = +1 \\ \boldsymbol{w} \cdot \boldsymbol{x}_i + b < 0 & \text{for } y_i = -1 \end{cases} \tag{2.22}$$

In the next step the data should not only be separated by a hyperplane but by a so called 'fat plane' [102], i.e. there should be a margin M between the hyperplane and the data points closest to it (in fig. 2.4(a) this is marked by the grey area surrounding the hyperplane A). Eqn.'s 2.22 are then modified to incorporate the margin [41].

$$\begin{cases} \boldsymbol{w} \cdot \boldsymbol{x}_i + b \geq 1 & \text{for } y_i = +1 \\ \boldsymbol{w} \cdot \boldsymbol{x}_i + b \leq -1 & \text{for } y_i = -1 \end{cases} \tag{2.23}$$

The above equations can be merged into one [41].

$$y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b \right) \geq 1 \quad \forall\, i \tag{2.24}$$

It can easily be shown that the size of the margin M is

$$M = \frac{2}{\|\boldsymbol{w}\|_2} \tag{2.25}$$

If the data is linearly separable then in general there is more than one hyperplane fulfilling eqn. 2.24. The goal is to find the optimal hyperplane such that a maximum margin is obtained (marked with O in the example fig. 2.4(a)) [22]. Those data points bounding the margin and which are required to describe the hyperplane exactly are called support vectors. Finding this hyperplane is a quadratic programming problem [102]. To be more precise it is a minimisation with constraint since the inverse of the margin (and not the margin itself) is used for optimisation [106]. The task reads as follows:

$$\begin{cases} \text{minimise :} & \frac{1}{2} \|\boldsymbol{w}\|_2^2 \\ \text{constraint :} & y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b \right) \geq 1 \quad \forall\, i \end{cases} \tag{2.26}$$

In many cases the original data cannot be separated entirely although the overall data structure warrants linear separation. In these circumstances the constraint (eqn. 2.24) should be relaxed to allow some data points, but only a small number, to violate it [102]. Hence, in the end some data points may lie within the margin or might even be situated on the "wrong" side of the decision hyperplane. To achieve this positive slack variables ξ_i are introduced [41]. The optimisation task (eqn.'s 2.26) is then slightly modified:

$$\begin{cases} \text{minimise :} & \frac{1}{2} \|\boldsymbol{w}\|_2^2 + \lambda \sum\limits_i \xi_i \\ \text{constraint 1 :} & y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b \right) \geq 1 - \xi_i \quad \forall\, i \\ \text{constraint 2 :} & \xi_i \geq 0 \quad \forall\, i \end{cases} \tag{2.27}$$

where λ is a parameter depending on how strong errors should be penalised. λ can be chosen as required. Larger values will result in stronger error penalty [102].

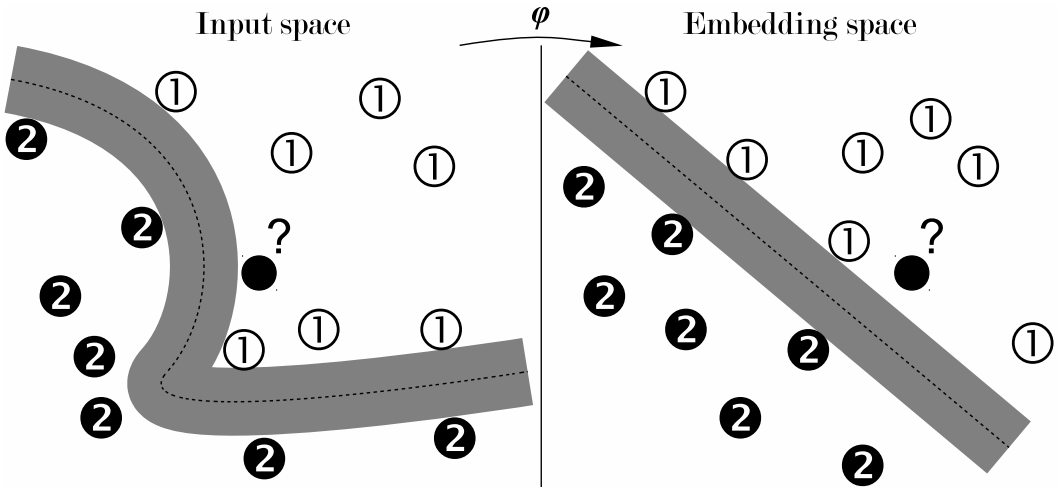


Figure 2.5: Mapping linearly non-seperable data to embedding space via $\boldsymbol{\varphi}$.

Support vector machines work with linear separation of data. If the data structure suggests non-linear separation (see fig. 2.4(b) for an example) a mapping $\boldsymbol{\varphi}(\boldsymbol{x})$ is applied to the data in input space such that is becomes linearly separable in an embedding space (fig. 2.5). The optimisation is therefore carried out in embedding space as well. Therefore eqn.'s 2.26 become

$$\begin{cases} \text{minimise :} & \frac{1}{2}\left\|\boldsymbol{W}\right\|_2^2 + \lambda\sum\limits_i \Xi_i \\ \text{constraint 1 :} & y_i\left(\boldsymbol{W}\cdot\boldsymbol{\varphi}\left(\boldsymbol{x}_i\right) + B\right) \geq 1 - \Xi_i \quad \forall\, i \\ \text{constraint 2 :} & \Xi_i \geq 0 \quad \forall\, i \end{cases} \tag{2.28}$$

The embedding space is usually of a much higher dimension than the original input space. In fact, the embedding space often has a dimensionality in the range of a million or more [102]. Solving this quadratic programming problem (eqn.'s 2.28) with a regular computer is not feasible. However, instead of looking at the primal problem one can look at the equivalent dual problem [106]:

$$\begin{cases} \text{minimise :} & \frac{1}{2}\boldsymbol{\alpha}\cdot\operatorname{diag}\left(\boldsymbol{y}\right)\cdot\boldsymbol{K}\cdot\operatorname{diag}\left(\boldsymbol{y}\right)\cdot\boldsymbol{\alpha} - \mathbf{e}\cdot\boldsymbol{\alpha} \\ \text{constraint 1 :} & 0 \leq \alpha_i \leq \lambda \quad \forall\, i \\ \text{constraint 2 :} & \boldsymbol{\alpha}\cdot\boldsymbol{y} = 0 \end{cases} \tag{2.29}$$

where α_i are Lagrange multipliers, $\boldsymbol{e}$ is a unit vector, and $\boldsymbol{K}$ is the *kernel* with

$$K_{ij} = K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \boldsymbol{\varphi}\left(\boldsymbol{x}_i\right)\cdot\boldsymbol{\varphi}\left(\boldsymbol{x}_j\right) \tag{2.30}$$

The problem is now much easier to handle as only all K_{ij} need to be calculated from mapping $\boldsymbol{\varphi}(\boldsymbol{x})$. As it turns out this can be simplified even further. It is not necessary to determine $\boldsymbol{\varphi}(\boldsymbol{x})$. Only the K_{ij} are required. These just need to be of a form that might originate from some mapping $\boldsymbol{\varphi}(\boldsymbol{x})$. This is the so-called *kernel trick* [26, 41, 106]. Properties of the kernel are [102]:

- K_{ij} is symmetric in i and j

- The kernel has non-negative eigenvalues
- A multinomial combination of kernels also gives a kernel
- $K\left(\boldsymbol{\varphi}\left(\boldsymbol{x}_i\right), \boldsymbol{\varphi}\left(\boldsymbol{x}_j\right)\right)$ is a kernel $\forall\, \boldsymbol{\varphi}\left(\boldsymbol{x}\right)$ if $K\left(\cdot, \cdot\right)$ itself is one
- $K_{ij} = \boldsymbol{f}\left(\boldsymbol{x}_i\right) \cdot \boldsymbol{f}\left(\boldsymbol{x}_j\right)$ is a kernel $\forall\, \boldsymbol{f}\left(\boldsymbol{x}\right)$

Furthermore, one is not required to find the kernel for a specific problem. In many cases a standard kernel can be used instead with excellent results. Typical kernels are [60, 106]:

$$K_{ij} = \begin{cases} \boldsymbol{x}_i \cdot \boldsymbol{x}_j & \text{linear} \\ \left(\gamma \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \lambda\right)^d & \text{polynomial} \\ e^{-\gamma \left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|^2} & \text{RBF} \\ \tanh\left(\gamma \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \lambda\right) & \text{sigmoid} \end{cases} \tag{2.31}$$

where γ and d are parameters.

Up to know the use of support vector machines for classification was dicussed. But SVM are also applicable to regression problems. Support vector regression is very similar to classification. First, the following definitions are made [43]:

Let ε be a tube around the prediction "curve" of the support vector machine. Then ξ_i is the positive difference between the observed value and the tube if the observed value is "above" the prediction tube. If the observed value is inside the tube or "below" the tube $\xi_i = 0$. One also defines ξ_i^* as the positive difference between the observed value and the tube if the observed value is "below" the prediction tube. If the observed value is inside the tube or "above" the tube $\xi_i^* = 0$. Fig. 2.6 visualises the definitions.

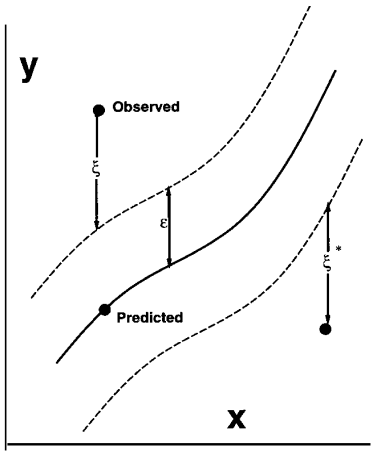


Figure 2.6: The parameters for support vector regression [43].

Then the optimisation task (eqn.'s 2.28) can be written for regression as [43]

$$\begin{cases} \text{minimise :} & \frac{1}{2}\left\|\boldsymbol{W}\right\|_2^2 + \lambda\sum\limits_i \Xi_i + \lambda\sum\limits_i \Xi_i^* \\ \text{constraint 1 :} & y_i - \boldsymbol{W}\cdot\boldsymbol{\varphi}\left(\boldsymbol{x}_i\right) - B \leq \varepsilon + \Xi_i \quad \forall\, i \\ \text{constraint 2 :} & \boldsymbol{W}\cdot\boldsymbol{\varphi}\left(\boldsymbol{x}_i\right) + B - y_i \leq \varepsilon + \Xi_i^* \quad \forall\, i \\ \text{constraint 3 :} & \xi_i, \xi_i^* \geq 0 \quad \forall\, i \end{cases} \tag{2.32}$$

The transformation into the dual form including the kernel trick is done as for classification:

$$\begin{cases} \text{minimise :} & \frac{1}{2}\left(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}\right)\cdot\boldsymbol{K}\cdot\left(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}\right) + \varepsilon\mathbf{e}\cdot\left(\boldsymbol{\alpha}^* + \boldsymbol{\alpha}\right) - \boldsymbol{y}\cdot\left(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}\right) \\ \text{constraint 1 :} & 0 \leq \alpha_i \leq \lambda \quad \forall\, i \\ \text{constraint 2 :} & 0 \leq \alpha_i^* \leq \lambda \quad \forall\, i \\ \text{constraint 3 :} & \mathbf{e}\cdot\boldsymbol{\alpha} = \mathbf{e}\cdot\boldsymbol{\alpha}^* \end{cases} \tag{2.33}$$

2.1.5 Bayesian Networks

Bayesian networks (BN) are probabilistic graphical models derived from Bayesian probability theory[11] [73]. A Bayesian net describes how different states x_i of a system are linked through probability, i.e. the net shows conditional inter-dependencies of variables via a directed acyclic graph [19].

In the graph each node represents one state of the system as a random variable. The edges show direct probabilistic relations between those states [51]. As the graph is directed, each state is only dependent on the parent states. Hence, for each node x_i there is a conditional probability distribution

$$P_B\left(x_i|\pi_i\right) \tag{2.34}$$

where π_i are the parent nodes of x_i. A unique joint probability distribution over the graph is then [73]:

$$P_B\left(x_1, x_2, \ldots, x_n\right) = \prod_{i=1}^{n} P_B\left(x_i|\pi_i\right) \tag{2.35}$$

Fig. 2.7 shows an example of a Bayesian network. Here, one can clearly see the inter-dependencies between variables as well as the conditional probabilities.

Bayesian networks can be learned from available data. A network must be found that represents the given training data as best as possible. This is extremely difficult [34]. However, with some restrictions on the networks, efficient procedures are available for teaching. These are separated into two categories [39, 59]:

- Structure learning (structure is unknown)
- Parameter learning (structure is known)

[11] T. Bayes 1763

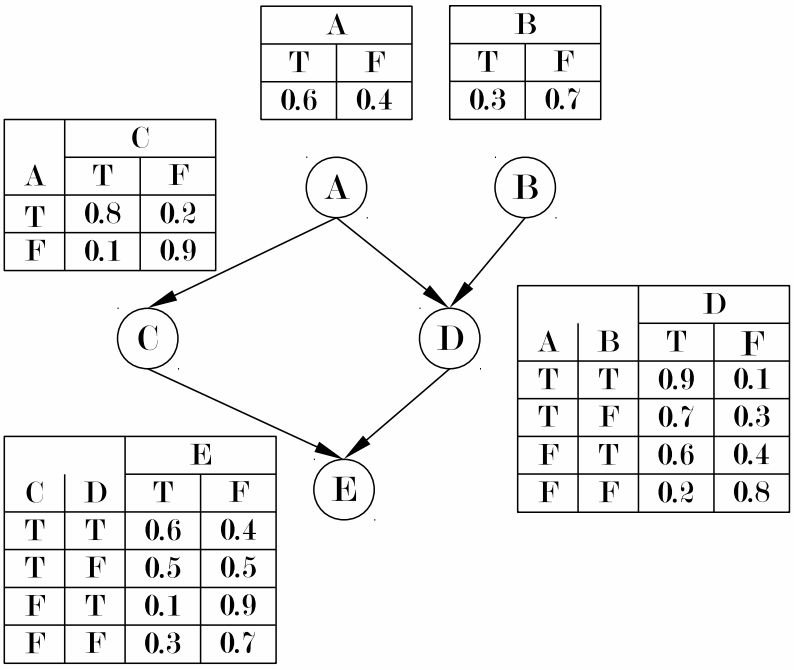


Figure 2.7: Bayesian network example.

Usually, parameter learning and structure learning are combined such that parameter learning is a sub-process of structure learning in a so called *score-and-search-based approach*. There are mainly two methods in use which are asympotically equivalent for large sample sizes [51]:

- Bayesian scoring [39, 59]
- Minimal description length (MDL) [52, 77, 112]

Here, only MDL is briefly discussed. Let B be a Bayesian network and $T = \{\boldsymbol{u}_1, \boldsymbol{u_2}, \ldots, \boldsymbol{u}_N\}$ a set of training data. Then one can write the *log likelyhood* of B given T as

$$LL\left(B|T\right) = \sum_{i=1}^{N} \log\left(P_B\left(\boldsymbol{u}_i\right)\right) \tag{2.36}$$

The scoring function of the Bayesian network reads

$$MDL\left(B|T\right) = \frac{\log N}{2}\left|B\right| - LL\left(B|T\right) \tag{2.37}$$

where $|B|$ is the number of parameters in B [51]. Maximising eqn. 2.36 will give a network that represents the given training data best. However, this results in very large networks and overfitting. To optimise the network also in the sense of its size and to avoid overfitting eqn. 2.37 is to be minimised as the size of the network is taken into account through $|B|$ [51].

2.1.6 Multiple Linear Regression

Multiple linear regression (MLR) is very similar to linear regression. But instead of having one independent variable as for linear regression output y depends on two or more (in general p) independent variables x_i, i.e. one has a relation of the form

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \tag{2.38}$$

where α and β_i are constants [46].

Given a sample set of size n one can write for $j = 1, \ldots, n$

$$y_j = \alpha_j + \beta_1 x_{j1} + \beta_2 x_{j2} + \cdots + \beta_p x_{jp} \tag{2.39}$$

Define the vector of the dependent variable $\boldsymbol{y}$, the deviation vector $\boldsymbol{\alpha}$, the coefficient vector $\boldsymbol{\beta}$, and the data matrix of the independent variables as follows:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \text{ and } \boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \tag{2.40}$$

Then, eqn. 2.39 can be written for all j as a system of equations.

$$\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{X}\boldsymbol{\beta} \tag{2.41}$$

Learning a multiple linear regression machine is done via the method of least squares [60]. For this purpose, one has to find $\boldsymbol{b}$ such that $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2$ becomes minimal. According to the *Gauss-Markov theorem* the solution is the best linear unbiased estimator with

$$\boldsymbol{b} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{2.42}$$

Therefore, one can rewrite eqn. 2.41 as

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{e} \tag{2.43}$$

with the remainder $\boldsymbol{e}$ and $\hat{\boldsymbol{y}}$ the estimated value of $\boldsymbol{y}$:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{b} \tag{2.44}$$

This way one obtains an equation for the estimated value $\hat{y}$ of the dependent variable y:

$$\hat{y} = b_1 x_1 + b_2 x_2 + \cdots + b_p x_p \tag{2.45}$$

2.1.7 Combining Different Machine Learning Methods

Combining multiple machine learning models shows a much lower error compared to using only a single machine learning method [10]. The combination can be done in various ways. Perhaps the best known combination methods are:

- Uniform voting / uniform average

- Weighted voting / weighted average
  - Likelihood combination
  - Bayesian combination
  - Distribution summation

With uniform voting it is assumed that all machine learning model have the same reliability [10]. For a given input an output is obtained for each model. Then, in case of classification, the resulting class determined by the whole system is the one which is obtained most often in the models. When regression is used the result is the average of the outputs from the models (uniform average).

Uniform voting can be extended to weighted voting where the influence of each model on the classification by the system depends on the error determined during training. Similarly, a weighted average can be used for regression. The determination of the weights can be more sophisticated when using one of the methods briefly described below.

In likelihood combination the *degree of logical sufficiency of the rule* [44] is determined for each classification rule. From this the rule with the highest degree is choses and then multiplied by the a priori propability of the classification [10]. The resulting value (obtained for each model separately) is a measure for the reliability and can be used as a weight for the voting method.

For Bayesian combination [25] the accuracy of each rule is determined for each class. Given a set of samples the accuracy is the ratio between the number of samples of a class satisfying the rule and the total number of samples in the set. Then for each class the most reliable rule is determined and the classes are compared [10]. The highest accuracy yields the classification of the machine.

The distribution summation [36] produces a vector by summing vectors (one for each classification rule) that contain information on the number of samples covered by each rule [10]. The number of entries for each vector is equal to the the number of classes. The largest component in the resulting vector determines the classification.

2.1.8 Software Implementation

The machine learing algorithms described above are implemented in EIDOminer[12]. This software was originally developed for the analysis of casting processes. The application in this thesis is the first outside this scope and is done in close co-operation with the software developers to test and expand the capabilities of the software.

The software centres around an Intelligent Analysis Manager which allows the identification of causal relations between error patterns and process parameters [45]. It also allows the combination of different methods as described in section 2.1.7 through weighted voting or weighted average. Furthermore, the raw data can be pre-processed before passed to the learning machines. The following smoothing methods are implemented:

- Fast Fourier transform (FFT)

[12] EIDOdata GmbH (http://www.eidodata.de/)

- Convolution (Conv)
- Moving average (MA)
- Singular value decomposition (SVD)

In EIDOminer the FFT smoother eliminates frequency noise by setting all frequencies to zero that are less than a threshold times the maximum distance of the data. The convolution smoother uses the Gaussian kernel. Moving averages is done with 5 consecutive entries. And the SVD smoother removes singular values which are below a chosen threshold.

2.2 CFD Models

2.2.1 CAD Model

For the CFD analysis which is done with ANSYS CFX 15.0 a torpedo-shaped AUV body is constructed. This shape is chosen as it is well-understood and most autonomous underwater vehicles have this type of body. The main body is constructed from two ellipsoids and a cylinder (fig. 2.8). The cross-shaped ruder construction is also very typical. The CAD model is created with PTC Creo Parametric 2.0. Fig. 2.9 shows a 3-D view of the finished model.

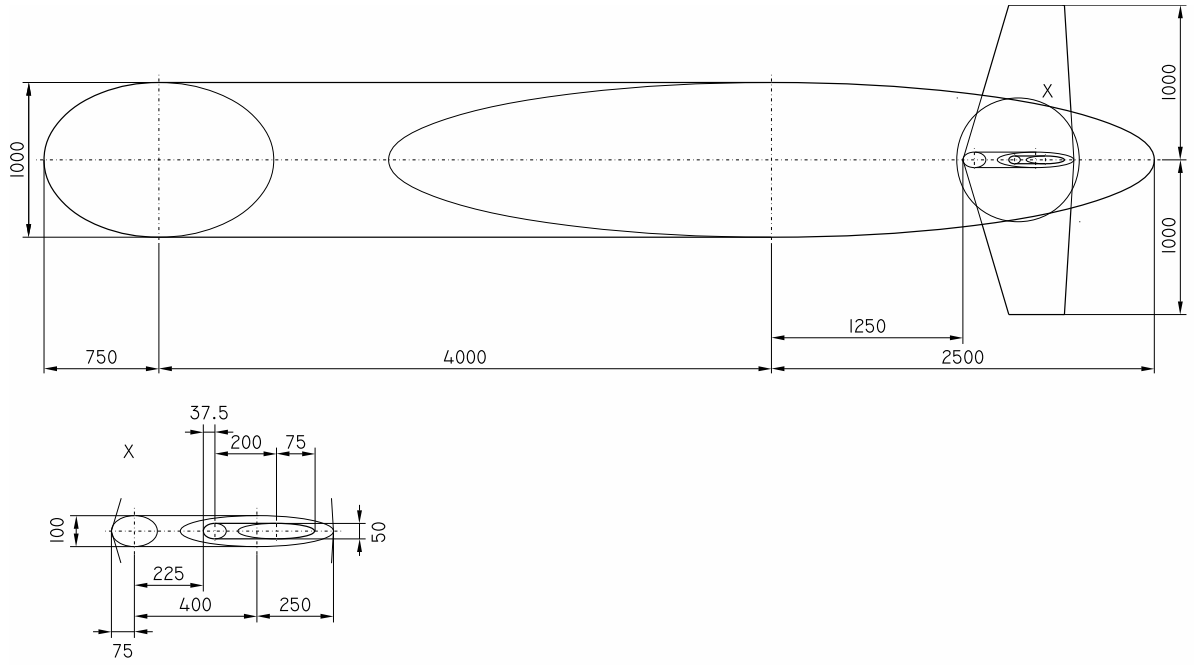


Figure 2.8: Blueprint of the AUV model for the CFD simulation.

The size of the model is quite large ($LOA = 7250\,mm$[13]). The reason is that pressure varies stronger than for small bodies. Especially, the pressure difference between the upper and lower side due to buoyancy is large enough (about $10\,000\,Pa$ per $1000\,mm$) to be significant. This way all influences can be considered in more detail and with better accuracy.

A fluid body with a length of $16\,000\,mm$ and $6000\,mm$ in the other two coordinate directions is created around the AUV model such that the forward boundary is $2750\,mm$ in front of the nose

[13]LOA: length overall

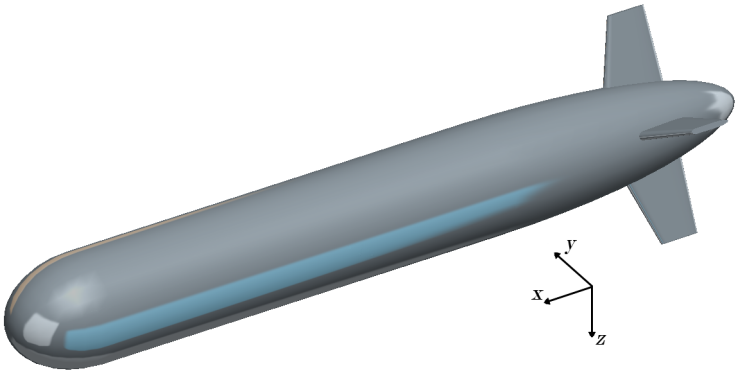


Figure 2.9: 3-D view of the AUV model for the CFD simulation.

of the AUV and the setup is symmetric in the other two coordinate directions (see fig. 2.10). A preliminary analysis with maximum velocities shows that a fluid body of this size is sufficient. The flow can fully develop before it comes in contact with the AUV model and all influences of the AUV on the fluid flow have dissipated before reaching the opposite boundaries.

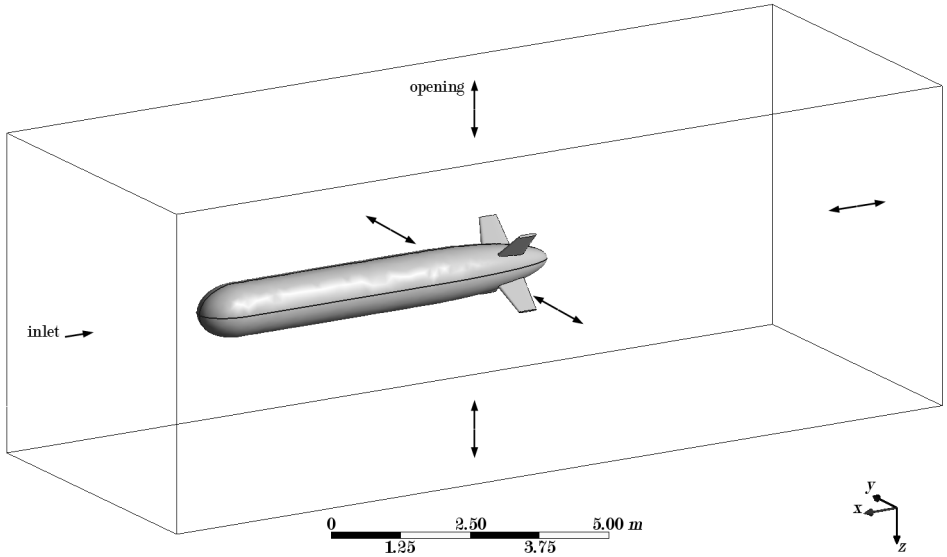


Figure 2.10: Fluid body for CFD simulation with boundaries for one flow situation.

2.2.2 Meshing

Mesh Parameters

A standard unstructured mesh is used for the simulations[14]. Tab. 2.2 shows some of the parameters of the mesh. Additionally, 5 prism layers are created on the surface of the AUV to resolve the transition layer. The transition ratio is 0.77 and the growth rate is 1.20. Fig. 2.11 shows a vertical cut through the final mesh.

Mesh Statistics and Mesh Quality

The mesh obtained has a total number of 92819 nodes and 525857 linear tetrahedral elements (Tet4). The average quality of the elements in the mesh is 0.83122 which is considered to be

[14]The mesh sensitivity analysis is discussed in section 2.2.7.

Curvature normal angle	18.0°
Minimum size	9.02710 mm
Maximum face size	902.710 mm
Maximum size	1805.40 mm
Growth rate	1.20

Table 2.2: Mesh parameters.

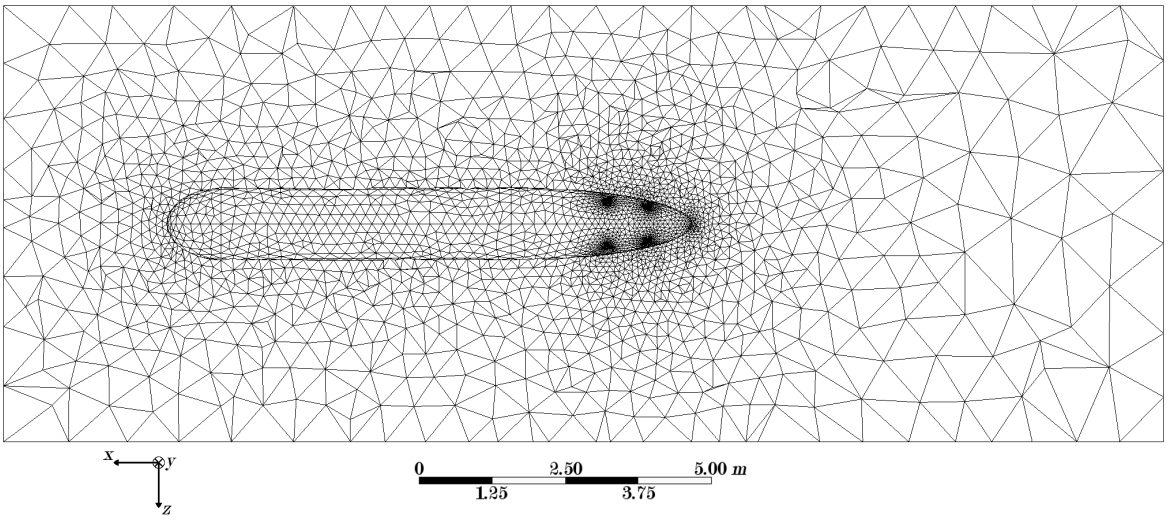


Figure 2.11: Mesh cut in the zx-plane.

excellent [13]. Other indicators of mesh quality, namly aspect ratio, maximum corner angle, skewness, and orthogonal quality, have excellent values as well[15].

2.2.3 Physical Properties of Seawater

As seawater is not a standard material and therefore usually not implemented in CFD software the necessary physical properties have to be calculated and implemented in the simulation. The relevant properties are

- Salinty S
- Molar mass M
- Density ρ
- Specific heat capacity c
- Thermal conductivity k
- Dynamic viscosity μ
- Specific enthalpy h
- Specific entropy s
- Thermal expansion coefficient α

[15]More details can be found in the appendix (section 7.1). Definitions of the indicators can be found in [13].

Salinity

Seawater is a solution of various solutes ("sea salt") in water. The composition of the solutes is very uniform through the oceans [5]. However, the amount of sea salt dissoved in water varies depending on the location. On average the mass fraction of solutes, called *salinity*, is about $S = 35\,g\,kg^{-1}$[16] which will be used for the purpose of this investigation. All physical properties of seawater are a function of salinity as can be seen below.

Molar Mass

For a mixture an average molar mass M_{mix} can be calculated from the molality b_i and the molar masses M_i of its constituents.

$$M_{mix} = \sum_i \frac{b_i}{b_{tot}} M_i \tag{2.46}$$

with

$$b_{tot} = \sum_i b_i \tag{2.47}$$

Tab. 2.3 shows the abundance of the major elements and molecules in seawater for a salinity of $S = 35\,g\,kg^{-1}$ and their respective molar masses. These make up most of the seawater. Other elements and molecules are of little consequence for the determination of the average molar mass as only trace amount are present [100].

Molecule / Element	**Molality** $[mol\,kg^{-1}]$	**Molar Mass** $[kg\,kmol^{-1}]$
H_2O	53.6	18.0153
Cl^-	0.5459	35.4530
Na^+	0.4690	22.9898
Br	0.1300	79.9040
Mg^{2+}	0.05283	24.3005
$[SO_4]^{2-}$	0.02823	96.0700
Ca^{2+}	0.01028	40.0780
K^+	0.01021	39.0983
$[HCO_3]^-$	0.00206	61.0168
Sr^{2+}	0.000091	87.6200

Table 2.3: Abundance of major elements and molecules in average seawater [100].

With the data from tab. 2.3 and eqn. 2.46 the average molar mass of seawater M_{sw} at a salinity of $S = 35\,g\,kg^{-1}$ can be determined as

$$M_{sw} = 18.4341\,kg\,kmol^{-1} \tag{2.48}$$

[16] The exact amount of normal salinity is $S_n = 35.16504\,g\,kg^{-1}$ [91].

Density

There are a number of empirical formulae to determine the density of seawater ρ_{sw} for given conditions. An overview can be found in [109]. Here the formula by Millero and Poisson [90] is used as the values obtained have an accuracy of $\pm 0.01\,\%$ which is higher than for most other formulae [109]. It is also valid in the relevant regions of temperature ($-2\,°C \leq T \leq 40\,°C$) and salinity ($0\,g\,kg^{-1} \leq S \leq 42\,g\,kg^{-1}$). The formula reads

$$\rho_{sw}\left[kg\,m^{-3}\right] = \rho_w + AS + BS^{3/2} + CS^2 \tag{2.49}$$

with the pre-factors

$$\begin{aligned} A &= 0.824493 - 4.0899 \cdot 10^{-3}T + 7.6438 \cdot 10^{-5}T^2 - 8.2467 \cdot 10^{-7}T^3 \cdots \\ &\quad \cdots + 5.3875 \cdot 10^{-9}T^4 \qquad (2.50) \\ B &= -5.72466 \cdot 10^{-3} + 1.0227 \cdot 10^{-4}T - 1.6546 \cdot 10^{-6}T^2 \qquad (2.51) \\ C &= 4.8314 \cdot 10^{-4} \qquad (2.52) \end{aligned}$$

and the density of pure water

$$\begin{aligned} \rho_w\left[kg\,m^{-3}\right] &= 999.842594 + 6.793952 \cdot 10^{-2}T - 9.09529 \cdot 10^{-3}T^2 \cdots \\ &\quad \cdots + 1.001685 \cdot 10^{-4}T^3 - 1.120083 \cdot 10^{-6}T^4 + 6.536336 \cdot 10^{-9}T^5 \qquad (2.53) \end{aligned}$$

which gives

$$\rho_w = 997.048\,kg\,m^{-3} \tag{2.54}$$

The density can then be calculated for the reference temperature $T_{ref} = 25°C$ (standard reference temperature for material properties in ASYS CFX).

$$\rho_{sw} = 1023.34\,kg\,m^{-3} \tag{2.55}$$

Specific Heat Capacity

For the specific heat capacity the formula by Millero et al. [89] is used. It has a high accuracy of $\pm 0.01\,\%$ and is valid in a temperature range of $5\,°C \leq T \leq 35\,°C$ and for a mass fraction of chloride $0\,g\,kg^{-1} \leq w_{Cl^-} \leq 22\,g\,kg^{-1}$. The mass fraction of chloride is obtained from the data in tab. 2.3.

$$w_{Cl^-} = b_{Cl^-}M_{Cl^-} = 19.3538\,g\,kg^{-1} \tag{2.56}$$

Hence, the mass fraction of chloride is in the required range. The specific heat capacity for seawater can therefore be determined by

$$c_{sw}\left[J\,kg^{-1}K^{-1}\right] = c_w + Aw_{Cl^-} + Bw_{Cl^-}^{3/2} \tag{2.57}$$

with the pre-factors

$$A = -13.81 + 0.1938T - 0.0025T^2 \tag{2.58}$$

$$B = 0.43 - 0.0099T + 0.00013T^2 \tag{2.59}$$

and the specific heat capacity of pure water

$$\begin{aligned} c_w\left[J\,kg^{-1}K^{-1}\right] = {} & 4217.4 - 3.72T + 0.141T^2 - 2.654\cdot10^{-3}T^3\cdots \\ & \cdots + 2.093\cdot10^{-5}T^4 \end{aligned} \tag{2.60}$$

which yields

$$c_w = 4179.23\,J\,kg^{-1}K^{-1} \tag{2.61}$$

The specific heat capacity at the reference temperature $T_{ref} = 25°C$ is therefore

$$c_{sw} = 3997.94\,J\,kg^{-1}K^{-1} \tag{2.62}$$

Thermal conductivity

The thermal conductivity of seawater depends not only on its temperature and salinity but also on the pressure. However, the influence of pressure on the overall value is very small. The formulation by Castelli et al. [28] is specifically design for a salinity of $S = 35\,g\,kg^{-1}$. It is valid for a temperature range of $0\,°C \leq T \leq 30\,°C$ and a pressure range of $10^5\,Pa \leq p \leq 1.4\cdot10^8\,Pa$ with an accuracy of $\pm0.4\,\%$. The reference pressure for the material properties in ASYS CFX is $p_{ref} = 1\,atm = 1.01325\cdot10^5\,Pa$ and all pressures occurring in the simulation are above this pressure. The equation may therefore be used in this case.

$$k_{sw}\left[W\,m^{-1}\,K^{-1}\right] = 0.55286 + 3.4025\cdot10^{-10}p + 1.8364\cdot10^{-3}T - 3.3058\cdot10^{-7}T^3 \tag{2.63}$$

This results in a thermal conductivity of

$$k_{sw} = 0.608751\,W\,m^{-1}\,K^{-1} \tag{2.64}$$

Dynamic Viscosity

According to Isdale et al. [71] the dynamic viscosity of seawater can be calculated with an accuracy of $\pm0.5\,\%$ in a temperature range of $5\,°C \leq T \leq 25\,°C$ and a salinity of $0\,g\,kg^{-1} \leq S \leq 40\,g\,kg^{-1}$

using the following relation.

$$\mu_{sw}\left[kg\,m^{-1}s^{-1}\right] \;=\; \mu_w\left(1 \,+\, AS \,+\, BS^2\right) \tag{2.65}$$

with pre-factors

$$A \;=\; 1.474\cdot 10^{-3} \,+\, 1.5\cdot 10^{-5}T \,-\, 3.927\cdot 10^{-8}T^2 \tag{2.66}$$

$$B \;=\; 1.073\cdot 10^{-5} \,-\, 8.5\cdot 10^{-8}T \,+\, 2.230\cdot 10^{-10}T^2 \tag{2.67}$$

and the pure water viscosity μ_w according to Korosi and Fabuss [76]

$$\ln\left(\mu_w\left[kg\,m^{-1}s^{-1}\right]\right) \;=\; -10.7019 \,+\, \frac{604.129}{139.18 \,+\, T} \tag{2.68}$$

which results in

$$\mu_w \;=\; 0.891839\cdot 10^{-3}\,kg\,m^{-1}s^{-1} \tag{2.69}$$

Hence, the dynamic viscosity of seawater is

$$\mu_{sw} \;=\; 0.958341\cdot 10^{-3}\,kg\,m^{-1}s^{-1} \tag{2.70}$$

Specific Enthalpy

A very accurate (accuracy $\pm 0.5\,\%$) relation between specific enthalpy of seawater and its salinity and temperature has been found by Sharqawy et al. [109] making use of the Gibbs energy function as layed down by IAPWS[17] in 2008 [5]. The relation is valid for $10\,°C \;\leq\; T \;\leq\; 120\,°C$ and $0\,g\,kg^{-1} \;\leq\; S \;\leq\; 120\,g\,kg^{-1}$.

$$\begin{aligned} h_{sw}\left[J\,kg^{-1}\right] \;=\;& h_w \,-\, S\left(a_1 \,+\, a_2S \,+\, a_3S^2 \,+\, a_4S^3 \,+\, a_5T \,+\, a_6T^2 \cdots\right. \\ & \left. \cdots + a_7T^3 \,+\, a_8ST \,+\, a_9S^2T \,+\, a_{10}ST^2\right) \end{aligned} \tag{2.71}$$

with the pre-factors[18]

[17] The International Association for the Properties of Water and Steam.

[18] Note that the pre-factors have been adjusted as the original formulation is not in SI units.

$$a_1 = -2.348 \cdot 10^1 \tag{2.72}$$
$$a_2 = 3.151 \cdot 10^{-1} \tag{2.73}$$
$$a_3 = 2.803 \cdot 10^{-3} \tag{2.74}$$
$$a_4 = -1.446 \cdot 10^{-5} \tag{2.75}$$
$$a_5 = 7.826 \tag{2.76}$$
$$a_6 = -4.417 \cdot 10^{-2} \tag{2.77}$$
$$a_7 = 2.139 \cdot 10^{-4} \tag{2.78}$$
$$a_8 = -1.991 \cdot 10^{-2} \tag{2.79}$$
$$a_9 = 2.778 \cdot 10^{-5} \tag{2.80}$$
$$a_{10} = 9.778 \cdot 10^{-5} \tag{2.81}$$

and the specific enthalpy of pure water at $T_{ref} = 25\,^{\circ}C$

$$h_w \left[J\,kg^{-1}\right] = 141.355 + 4202.07T - 0.535T^2 + 0.004T^3 \tag{2.82}$$

which gives

$$h_w = 104921\,J\,kg^{-1} \tag{2.83}$$

The seawater specific enthalpy is therefore

$$h_{sw} = 99765.1\,J\,kg^{-1} \tag{2.84}$$

Specific Entropy

Like specific enthalpy, specific entropy can be determined from the Gibbs energy function. The relation by Sharqawy et al. [109] reduces the errors of previous formulations from up to $\pm 35\,\%$ to $\pm 0.5\,\%$. As for specific enthalpy, the relation is valid for $10\,^{\circ}C \leq T \leq 120\,^{\circ}C$ and $0\,g\,kg^{-1} \leq S \leq 120\,g\,kg^{-1}$.

$$\begin{aligned} s_{sw}\left[J\,kg^{-1}K^{-1}\right] = {} & s_w - S\left(a_1 + a_2 S + a_3 S^2 + a_4 S^3 + a_5 T + a_6 T^2 \cdots\right. \\ & \left. \cdots + a_7 T^3 + a_8 S T + a_9 S^2 T + a_{10} S T^2\right) \end{aligned} \tag{2.85}$$

with the pre-factors[19]

[19] Again the pre-factors have been adjusted due to non-SI units used in the original formulation.

$$a_1 = -4.231 \cdot 10^{-1} \tag{2.86}$$
$$a_2 = 1.463 \cdot 10^{-2} \tag{2.87}$$
$$a_3 = -9.880 \cdot 10^{-5} \tag{2.88}$$
$$a_4 = 3.095 \cdot 10^{-7} \tag{2.89}$$
$$a_5 = 2.562 \cdot 10^{-2} \tag{2.90}$$
$$a_6 = -1.443 \cdot 10^{-4} \tag{2.91}$$
$$a_7 = 5.879 \cdot 10^{-7} \tag{2.92}$$
$$a_8 = -6.111 \cdot 10^{-5} \tag{2.93}$$
$$a_9 = 8.041 \cdot 10^{-8} \tag{2.94}$$
$$a_{10} = 3.035 \cdot 10^{-7} \tag{2.95}$$

and the specific entropy of pure water at $T_{ref} = 25\,°C$

$$s_w \left[J\,kg^{-1}K^{-1}\right] = 0.1543 + 15.383T - 2.996 \cdot 10^{-2}T^2 + 8.193 \cdot 10^{-5}T^3 \cdots \\ \cdots - 1.370 \cdot 10^{-7}T^4 \tag{2.96}$$

which yields

$$s_w = 367.231\,J\,kg^{-1}K^{-1} \tag{2.97}$$

The resulting specific entropy is

$$s_{sw} = 349.860\,J\,kg^{-1}K^{-1} \tag{2.98}$$

Thermal Expansion coefficient

A formula for the relation between the thermal expansion coefficient and temperature, pressure, and salinity is not known. However, for certain combinations some values have been tabulated [113]. For a temperature of $T = 25\,°C$, a salinity of $S = 35\,g\,kg^{-1}$, and standard pressure the thermal expansion coefficient is

$$\alpha = 297 \cdot 10^{-6}\,K^{-1} \tag{2.99}$$

2.2.4 Fluid Model

For the simulations an isothermal buoyant k-ε model with scalable wall functions is chosen with a reference pressure of $p_{ref} = 2\,atm = 2.02650 \cdot 10^5\,Pa$, which corresponds to a diving depth of approximately $d_{dive} \approx 10\,m$, and a fluid temperature of $T = 15\,°C$. The reason is that no large adverse pressure gradients are expected and the k-ε model works especially well for pressure

gradients which are relatively small [15]. Furthermore, the control volume is large enough, such that gravity will result in a significant difference in static pressure between its upper and lower side. Hence, buoyancy has to be taken into account. In addition, a uniform temperature is set throughout the simulations. The k-ε model is also advantageous, as the number of simulations is quite large (1546 simulations in total). The k-ε model provides a reasonable compromise between accuracy and computational efford [120], such that the complete set of simulations can be run within an adequate time frame.

For a mathematical model of fluid flow the underlying principles are the conservation of mass, momentum, and energy [49, 107]. These principles are expressed through the following equations.

Continuity equation (conservation of mass):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{2.100}$$

Navier-Stokes equations (conservation of momentum):

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \mu \nabla^2 \boldsymbol{u} - \frac{2}{3}\mu \nabla (\nabla \cdot \boldsymbol{u})\,\delta + \rho \boldsymbol{g} \tag{2.101}$$

Energy equation (conservation of energy):

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \boldsymbol{u} h) = -\nabla \cdot q + \frac{\partial p}{\partial t} + \nabla \cdot (\tau \cdot \boldsymbol{u}) \tag{2.102}$$

When using the k-ε model in an isothermal situation the energy equation is not required. The reasons are that most terms in eqn. 2.102 become zero in this case and that the turbulence energy and its dissipation are included in the other equations (as shown below).

The k-ε model is a widely used two-equation turbulence model for fluid dynamical problems. It was introduced by Launder and Spalding in 1974 [78]. One defines the turbulent kinetic energy k as the variance of the fluctuations in velocity:

$$k = \frac{1}{2}\left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2}\right) \tag{2.103}$$

where u', v', and w' are the components of the velocity fluctuations $\boldsymbol{u}'$. Then ε is defined as the rate at which k dissipates:

$$\varepsilon = \frac{dk}{dt} \tag{2.104}$$

These variables are introduced in the continuity equation (eqn. 2.100) and the Navier-Stokes equations (eqn. 2.101). After some analytical manipulation one obtains the following equations (as implemented in ANSYS CFX [12]).

$$\frac{\partial (\rho k)}{\partial t} + \nabla \cdot (\rho k \boldsymbol{u}) = \nabla \cdot \left(\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right) + P_k + P_{kb} - \rho\varepsilon \tag{2.105}$$

$$\frac{\partial (\rho \varepsilon)}{\partial t} + \nabla \cdot (\rho \varepsilon \boldsymbol{u}) = \nabla \cdot \left(\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right) + \frac{\varepsilon}{k}\left(C_{\varepsilon 1} P_k - C_{\varepsilon 2}\rho\varepsilon + C_{\varepsilon 1} P_{\varepsilon b}\right) \tag{2.106}$$

with constants

$$C_{\varepsilon 1} = 1.44 \tag{2.107}$$
$$C_{\varepsilon 2} = 1.92 \tag{2.108}$$
$$\sigma_k = 1.00 \tag{2.109}$$
$$\sigma_\varepsilon = 1.30 \tag{2.110}$$

In these equations P_k is the turbulence production due to viscous forces with

$$P_k = \mu_t S^2 \tag{2.111}$$

where S is the modulus of the mean rate-of-strain tensor S_{ij}:

$$S = \sqrt{2 S_{ij} S_{ij}} \tag{2.112}$$

P_{kb} and $P_{\varepsilon b}$ represent the influence of the buoyancy forces. The buoyancy production term reads

$$P_{kb} = -\frac{\mu_t}{\rho \sigma_\rho} g \nabla \rho \tag{2.113}$$

with the turbulence Schmidt number

$$\sigma_\rho = 1.00 \tag{2.114}$$

It is assumed that $P_{\varepsilon b}$ is proportional to the buoyancy production and must be positive. Therefore

$$P_{\varepsilon b} = C_{\varepsilon 3} \cdot \max\left(0, P_{kb}\right) \tag{2.115}$$

with the dissipation coefficient

$$C_{\varepsilon 3} = 1.00 \tag{2.116}$$

Furthermore, the velocity scale ϑ and the length scale l are defined as

$$\vartheta = k^{1/2} \tag{2.117}$$

$$l = \frac{k^{3/2}}{\varepsilon} \tag{2.118}$$

such that the eddy viscosity μ_t can be written as

$$\mu_t = C_\mu \rho \vartheta l = \rho C_\mu \frac{k^2}{\varepsilon} \tag{2.119}$$

with constant

$$C_\mu = 0.09 \tag{2.120}$$

With the k-ε model Launder and Spalding also introduced wall functions [78]. It has been shown in experiments that in turbulent flows the velocity distribution closed to a wall is similar when regarded in a dimensionless form [107]. One defines the shear velocity u_τ through wall shear stress τ_w and density ρ as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{2.121}$$

The surface normal coordinate y and the mean surface parallel velocity $\bar{u}_t$ can than be de-dimensionalised to y^+ and u^+.

$$y^+ = \frac{y u_\tau}{\nu} \tag{2.122}$$

$$u^+ = \frac{\bar{u}_t}{u_\tau} \tag{2.123}$$

In the viscous sub-layer ($y^+ < 5$) the relation between u^+ and y^+ is roughly linear, i.e.

$$u^+ = y^+ \tag{2.124}$$

In the transition layer ($5 < y^+ < 70$) the functional relation is rather complex [105]. For $y^+ > 70$ the relation between u^+ and y^+ is

$$u^+ = \frac{1}{\kappa} \ln\left(y^+\right) + C_w \tag{2.125}$$

where κ is the Karman constant

$$\kappa = 0.41 \tag{2.126}$$

and C_w is the log-layer constant. It describes the influence of the surface roughness on the velocity distribution. For a smooth surface one has

$$C_w = 5.5 \tag{2.127}$$

Unfortunately, eqn. 2.125 has singularities at points close to the wall where $\bar{u}_t$ goes towards zero [12]. Hence, a different velocity scale u^* is introduced, replacing u_τ in the log-layer, which depends entirely on the turbulent kinetic energy and does not go to zero with $\bar{u}_t$ [12].

$$u^* = C_\mu^{1/4} k^{1/2} \tag{2.128}$$

Then one can also define a new dimensionless surface normal coodinate y^*.

$$y^* = \frac{y u^*}{\nu} \tag{2.129}$$

With these, shear velocity u_τ and the absolute value of the wall shear stress τ_w can be calculated with the following two equations.

$$u_\tau = \frac{\bar{u}_t}{\frac{1}{\kappa}\ln(y^*) + C} \tag{2.130}$$

$$\tau_w = \rho u^* u_\tau \tag{2.131}$$

Unfortunately, there are some significant problems when using wall functions with the above equations. According to Grotjans and Menter [56] the numerical results strongly depend on the location of the node cloest to the wall, i.e. the results are highly mesh dependent. A fine mech does not lead to a more accurate result. The solution may not even be unique. Hence, the wall functions by Launder and Spalding are extended to scalable wall functions [12]. In this approach, the value of y^* calculated in eqn. 2.129 is checked against a lower limit.

$$\tilde{y}^* = \max(y^*, 11.06) \tag{2.132}$$

Limiting $\tilde{y}^*$ according to eqn. 2.132[20] will result in all nodes lying outside the linear sub-layer in the logarithmic region such that inconsistencies due to fine meshing do not appear.

2.2.5 Boundary Conditions

Inlets

Up to three of the bounding surfaces of the control volume are defined as inlets, depending on the actual direction of the flow velocity vector $\boldsymbol{u}$. Tab. 2.4 shows the components of the inlet velocities chosen for the simulations. All possible combinations (except for $\boldsymbol{u} = [0,0,0]^T$) are used. However, as the system is symmetric in the zx-plane only positive v need to be evaluated. Then, the results for negative v are obtained by mirroring the results for positive v at the zx-plane. Hence, a set of 1546 simulations can be obtained.

u $[kn]$	20	15	10	7.5	5.0	2.5	2.0	1.5	1.0	0.5	0.0	-0.5	-1.0	-1.5	-2.0	-2.5	-5.0
v $[kn]$	-	-	-	-	5.0	2.5	2.0	1.5	1.0	0.5	0.0	-0.5	-1.0	-1.5	-2.0	-2.5	-5.0
w $[kn]$	-	-	-	-	5.0	2.5	2.0	1.5	1.0	0.5	0.0	-0.5	-1.0	-1.5	-2.0	-2.5	-5.0

Table 2.4: Components of the inlet velocity.

Furthermore, a medium turbulence intensity of $I = 0.05$ is assumed for the inlets. The turbulence intensity is defined as the fraction of velocity fluctuations with respect to the dominant flow velocity.

$$I = \frac{\boldsymbol{u}'}{\boldsymbol{u}} \tag{2.133}$$

The turbulence kinetic energy and its dissipation at the inlet can then be calculated [11]:

[20] At $y^* = 11.06$ the linear and the logarithmic region of the wall functions intersect.

$$k_{inlet} = \frac{3}{2} I^2 \boldsymbol{u}^2 \tag{2.134}$$

$$\varepsilon_{inlet} = \rho C_\mu \frac{k^2}{\mu_t} \tag{2.135}$$

As μ_t is defined by k and ε (see eqn. 2.119) one needs to obtain its value in a different way. In this special situation μ_t is calculated for the inlets as [11]

$$\mu_t = 1000 I \mu \tag{2.136}$$

Openings

The bounding surfaces of the control volume that are not inlets are modelled as openings with a relative pressure of $p_{rel} = 0$ normal to the boundary. This way the fluid can enter an leave the control volume freely as in open water. Furthermore, a relative pressure of $p_{rel} = 0$ ensures that no additional forces due to the boundary are acting on the fluid.

AUV Body

The AUV body is modelled as a smooth no slip wall, i.e. the fluid velocity at the wall is $\boldsymbol{u}_{wall} = 0$. Hence, there is no influence on the fluid flow due to surface roughness.

2.2.6 Simulation Parameters

As the conversion criterion for the calculations a root mean square error of 10^{-4} for mass, momentum, k, and ε is chosen. For the advection a high resolution scheme is used which is a nonlinear procedure based upon the boundedness principles according to Barth and Jesperson [16]. The turbulence numerics is also dealt with using a high resolution scheme as the standard first order approach runs into problems for some velocity vectors.

2.2.7 Sensitivity Analysis

A mesh sensitivity analysis is performed for several selected flow situations. It is shown that a finer mesh does not change the results of the simulations significantly. The maximum difference in the resulting pressures between the mesh used and the next finer standard mesh is less than 0.2 % which is sufficiently accurate for the given purpose.

2.2.8 Pressure Data Extraction and Processing

The pressure is extracted on the surface of the AUV at points 500 mm apart and situated on lines on the top, bottom, and the two sides of the AUV (see black lines on the surface of the AUV in fig. 2.10). The foremost points are 500 mm to the back of the bow of the AUV. This results in 14 points on each line and 56 points in total.

So for each flow situation there is data set consiting of 56 pressures and three flow velocity components. These are organised into a 2872×59 data matrix where each row represents one flow situations. The order of the rows is randomised. Randomisation is required to have a thorough mixture of matching pressures and velocity over the whole velocity range. The resulting data matrix is used as input for the machine learning software EIDOminer.

2.3 Image Processing

2.3.1 Laboratory Setup and Procedure

In order to obtain images under controlled lighting conditions various coloured object are illuminated by a special LED light source and with regular white light. The special LED light source has a range of $450\,nm$ to $570\,nm$ and a mean wavelength of $\bar{\lambda} = 498\,nm$ as can be seen in fig. 2.12. The mean wavelength is very close to the wavelength for maximum penetration depth in open ocean ($\lambda_{max,ocean} \approx 480\,nm$, see also fig. 1.7). Therefore, the objects look as if they where situated in deep ocean water and illuminated by a white light from a submarine or underwater robot from some distance away. In the following this is refered to as "underwater lighting conditions". For the regular white light a mercury-vapour lamp is used.

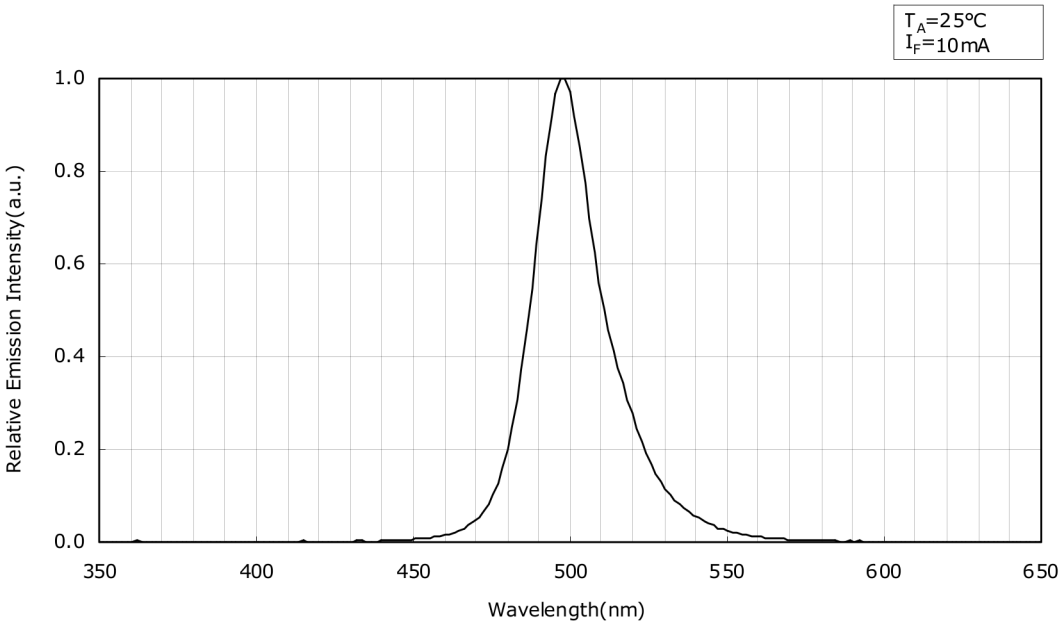


Figure 2.12: Spectrum of cyan LED [40].

The coloured objects are the photographed several times both under white light and under underwater lighting conditions. The camera used has an active pixel sensor (APS) with $16.1\,Mpx$ resolution. A sketch of the laboratory setup can be found in fig. 2.13.

2.3.2 Image Data Processing

The images taken are then processed as follows. First the images are resized such that only the coloured objects can be seen in the pictures. This is necessary as the background is not illuminated. Then a feature matching is done for each image set consisting of the same object photographed once under white light and once under underwater lighting conditions such that

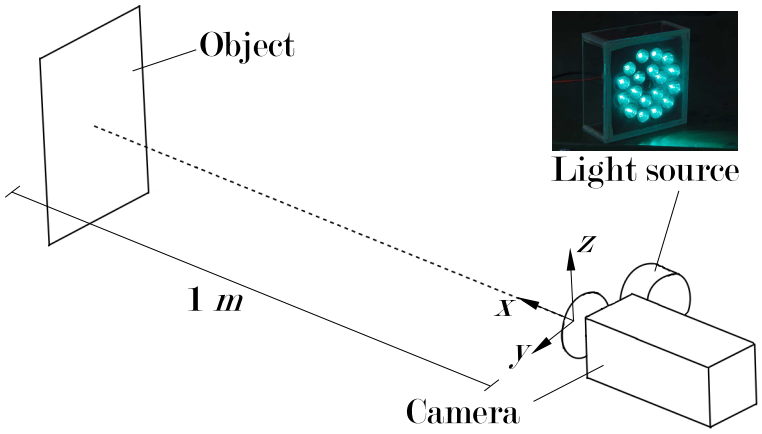


Figure 2.13: Laboratory setup for image acquisition.

the two pictures are exactly aligned. This way for every pixel in the image under underwater lighting conditions the corresponding pixel in the picture taken under white light is known.

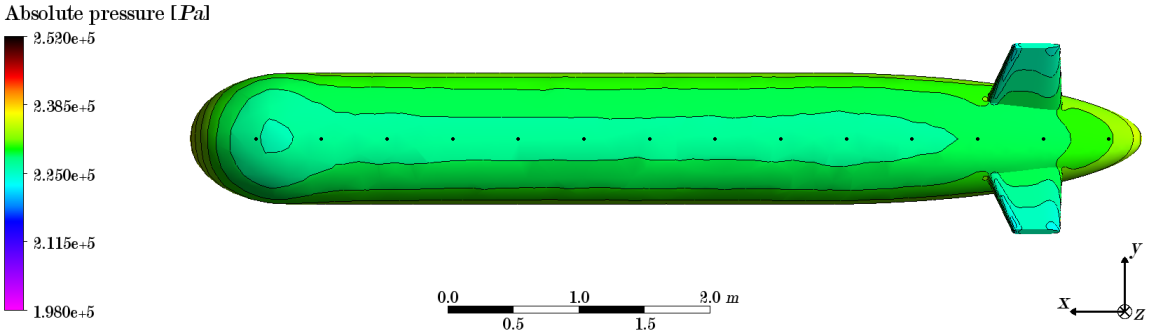
From every image set 10 % of the pixels are chosen at random and an $n \times 6$ matrix is built containing the colour channels (RGB) for two matching pixels in each row. These matrices are then merged into one data matrix and the order of the rows is randomised. Randomisation is required to have a thorough mixture of matching pixel data over the whole colour spectrum. The resulting data matrix is used as input for the machine learning software EIDOminer.

3 Results and Discussion

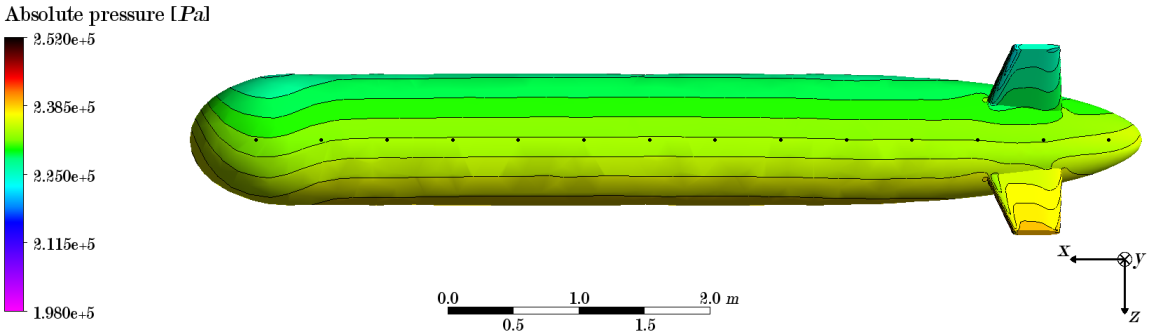
3.1 Underwater Navigation

3.1.1 CFD Simulation

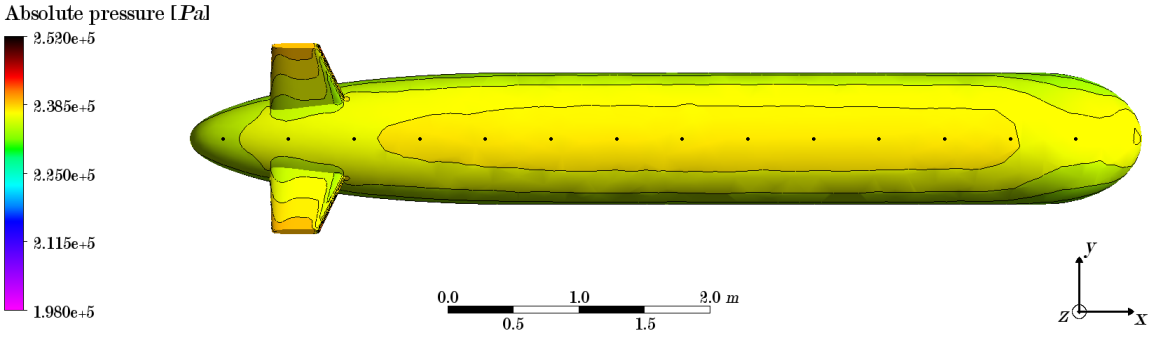
Fig. 3.1 shows the absolute pressure distribution on the hull of the AUV during forward motion at $5\,kn$ obtained from the CFD simulation. The small black dots indicate the positions of the pressure measurement points used for machine learning. The starboard view is not shown since it is a mirror image of the port view due to symmetry.



(a) Top view.



(b) Port view.



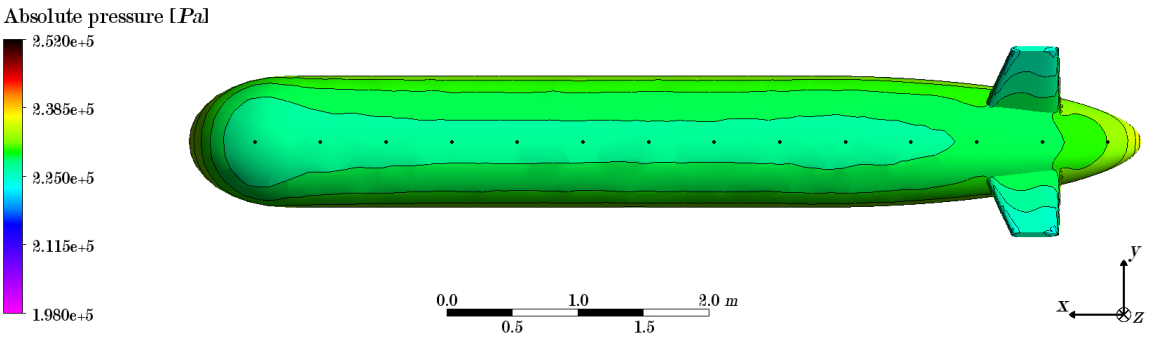
(c) Bottom view.

Figure 3.1: Forward motion at $5\,kn$.

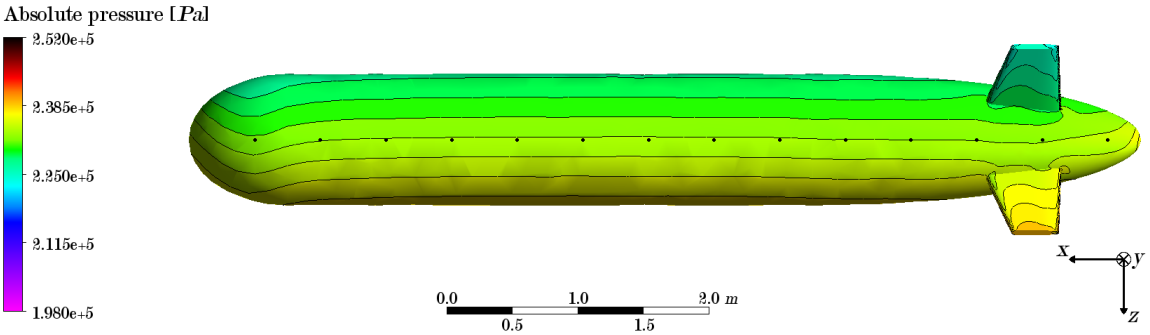
In the port view fig. 3.1(b) one can clearly observe the influence of buoyancy as the pressure increases strictly vertically with depth in the middle section of the AUV body. Only at the nose and the fins some influence due to the forward motion can be seen. A small low pressure region can be found at the top of the nose which is best seen in the top view fig. 3.1(a). At the same time the pressure is increased compared to the static pressure at the bottom of the nose (see right side of fig. 3.1(c)).

At the front of the fins the velocity head is observed. The pressure distribution then changes quickly such that it is in line with the distribution at the middle section of the AUV body. Toward the rear of the fin the pressure again changes but not as significantly as for the front. This behaviour is to be expected as it is typical for wing-type shapes in a fluid flow.

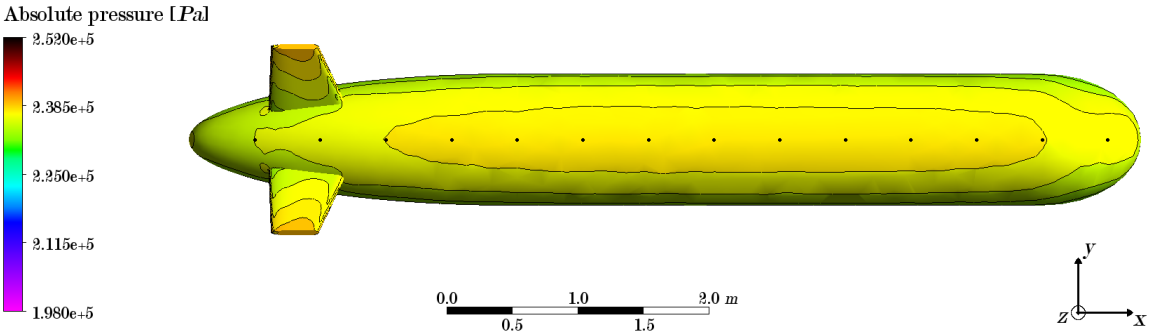
In fig. 3.2 the absolute pressure distribution during a backward motion at $5\,kn$ is shown. Again the starboard view is not given due to symmetry.



(a) Top view.



(b) Port view.



(c) Bottom view.

Figure 3.2: Backward motion at $5\,kn$.

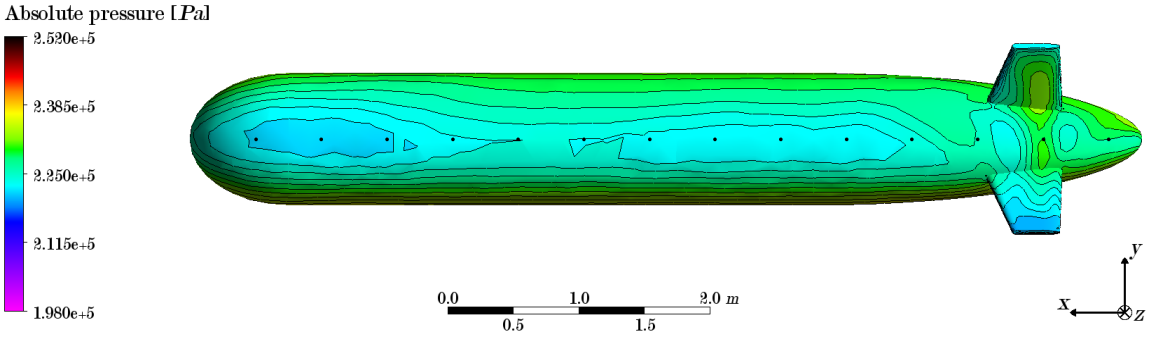
The results show only very small differences when compared to the forward motion. However, the small low pressure region at the top of the nose is no longer as profound (fig. 3.2(a)). Also the increase in pressure at the bottom of the nose is not as strong as for forward motion (fig. 3.2(c)).

The strongest influence of the backward motion can be observed at the back of the AUV body where a clear increase in pressure can be seen at the stern. This is shown best in the port view fig. 3.2(b) and the bottom view fig. 3.2(c).

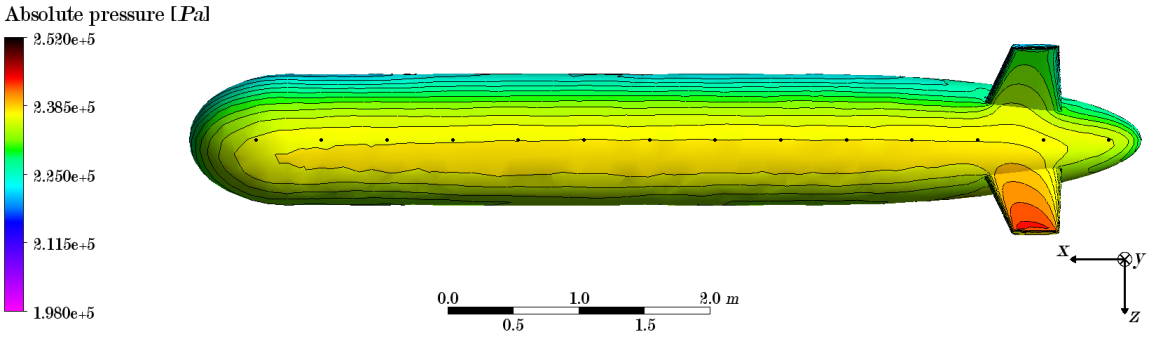
It is clear that both forward and backward motion do not disturbe the flow very much. The torpedo shape is very streamlined and optimised for fast forward motion. For the backward

motion the shape is not as optimal. Hence, the increased pressure at the back of the AUV body. However, the shape is still quite streamlined with respect to this particular flow situation.

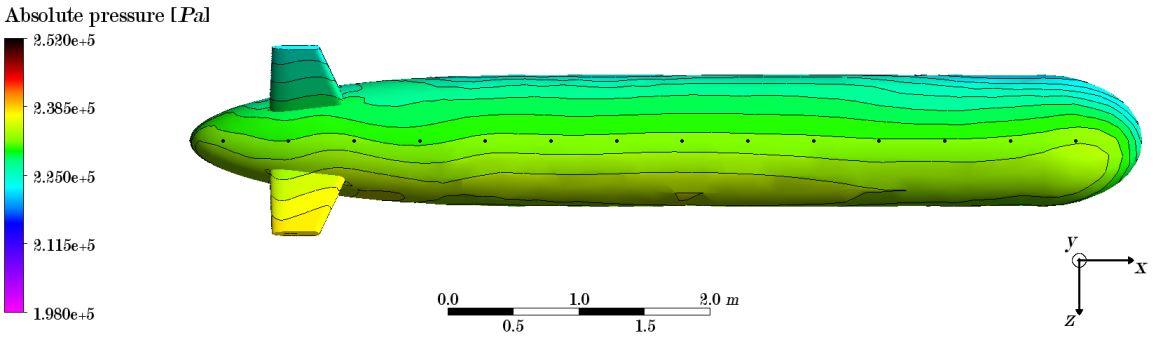
In the following figure (fig. 3.3) the absolute pressure distribution for a motion at 5 kn to port is shown as obtained from the CFD simulation. As there is no symmetry in this case also the starboard view is given. The results can also be used for the interpretation of a motion of 5 kn to starboard by mirroring the results at the z-x-plane.



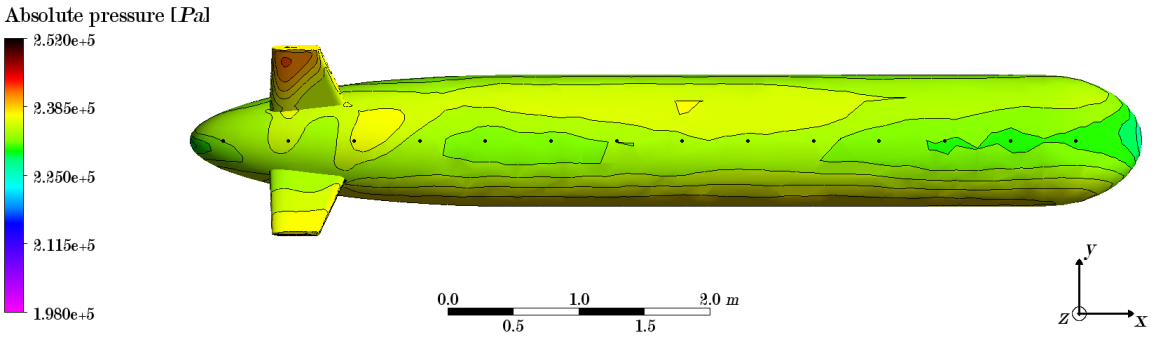
(a) Top view.



(b) Port view.



(c) Starboard view.



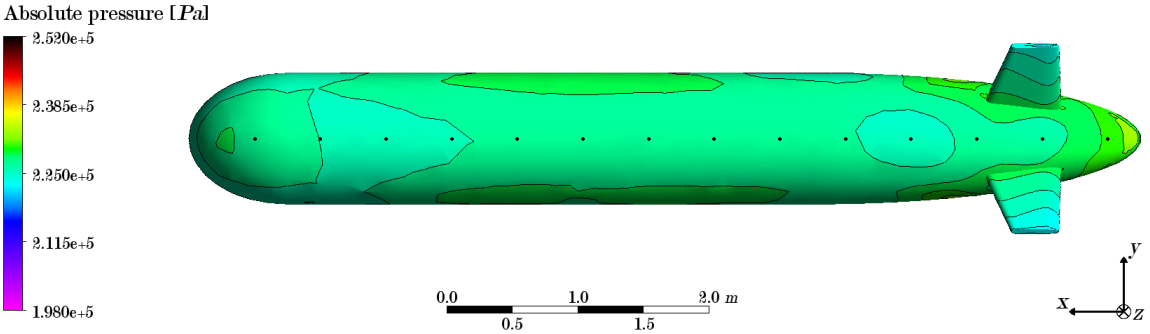
(d) Bottom view.

Figure 3.3: Motion to port at 5 kn.

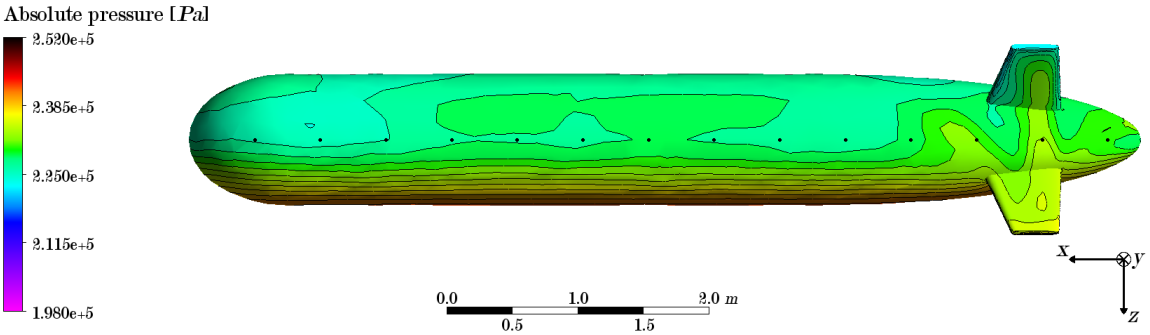
A high pressure region can be found almost over the whole length of the port side of the AUV body (fig. 3.3(b)). Almost all pressure measurement points on this side of the surface are within that region. Even higher pressures are observed at the outside of the lower port fin. A slight increase is also found at the upper port fin. However, the increase is not as strong.

On the starboard side fig. 3.3(c) away from the flow the pressure is slighlty reduced. While in the middle section the reduction is such that the pressure increase over the surface is still strictly horizontal, toward the back and front a wavy pattern can be observed. It is more pronounced at the back than at the front of the AUV body.

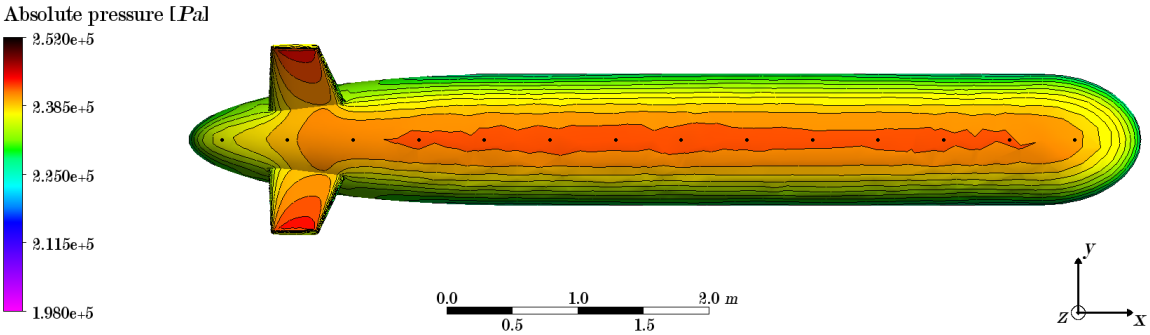
Looking at the top fig. 3.3(a) and bottom fig. 3.3(d) view one can clearly see that the port fins create strong turbulence at the surfaces toward starboard. The pressure changes rapidly between the fins and also the starboard fins are influenced by the turbulent region.



(a) Top view.



(b) Port view.



(c) Upward view.

Figure 3.4: Downward motion at 5 kn.

The shape of the AUV body is not optimal for the side flow situation. Also the surface of attack is much greater compared to forward or backward motion. Hence the pressure changes are much more pronounced in this case. Especially, the flat surfaces of the fins are directly in the way of the flow leading to a strong reduction of the local flow velocity and hence a high pressure.

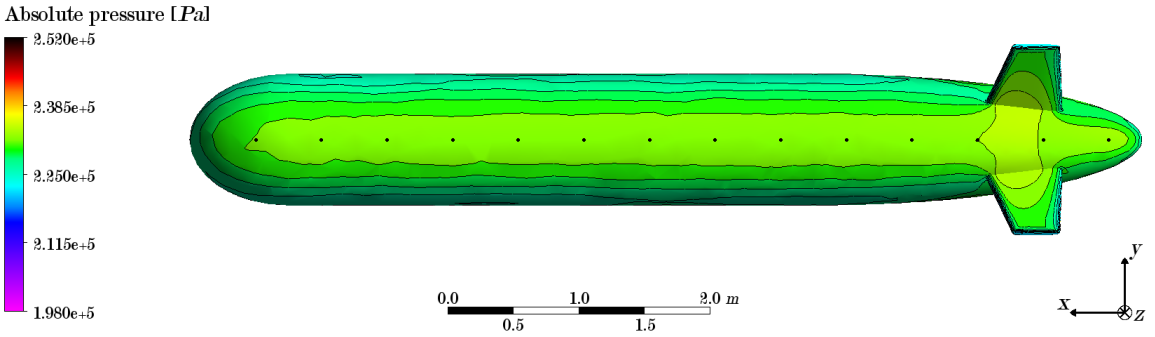
Fig. 3.4 shows the absolute pressure distribution on the hull of the AUV during downward motion at 5 kn. Since in this case symmetry is present again the starboard view is not given.

In case of downward motion a very distinct high pressure region along the bottom of the AUV hull is developed (fig. 3.4(c)). High pressures are also shown at the lower sides of the lower fins. Almost all pressure measurement points on the lower side are within the high pressure region.

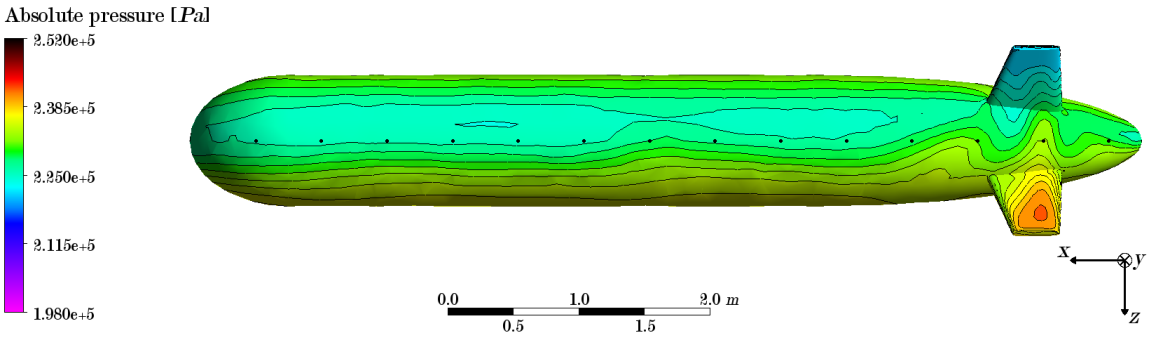
On the top side of the AUV body the pressure becomes very uniform with little variation (fig. 3.4(a)). The overall pressure is reduced compared to the static case.

Looking at the port view fig. 3.4(b) rapid changes in pressure are observed near the fins, i.e. a region of strong turbulence is created between the upper and the lower fins by the lower fins. There are strong qualitative similarities between this situation and the case of sideward motion.

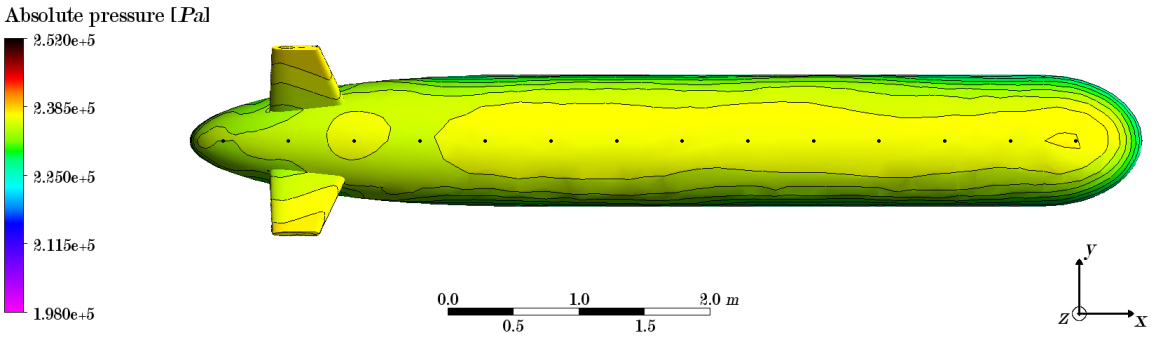
Again the shape of the AUV is not optimal for the given flow situation. The pressure on the lower side of the AUV is already high due to buoyancy and the additional downward motion creates even more pressure on the hull especially since the attack surface is large. As for the sideward flow situation the flat sides of the fins are directly in the way of the flow creating a low velocity / high pressure region toward the flow and turbulence on the side away from the flow.



(a) Top view.



(b) Port view.

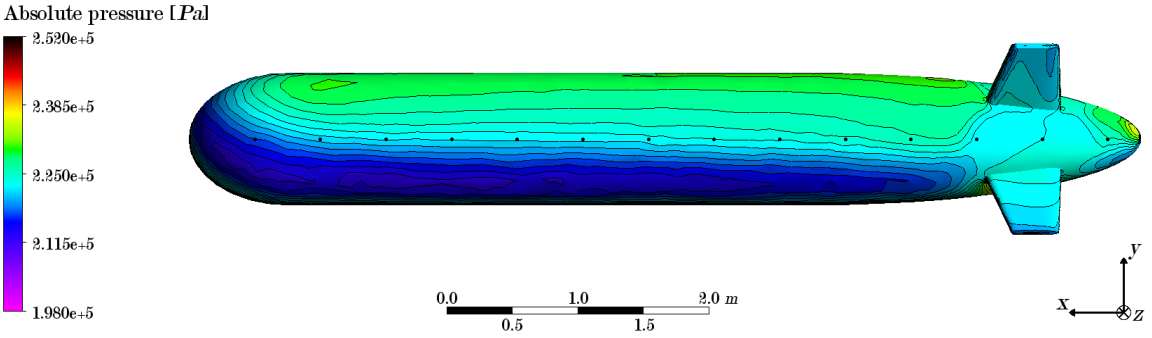


(c) Bottom view.

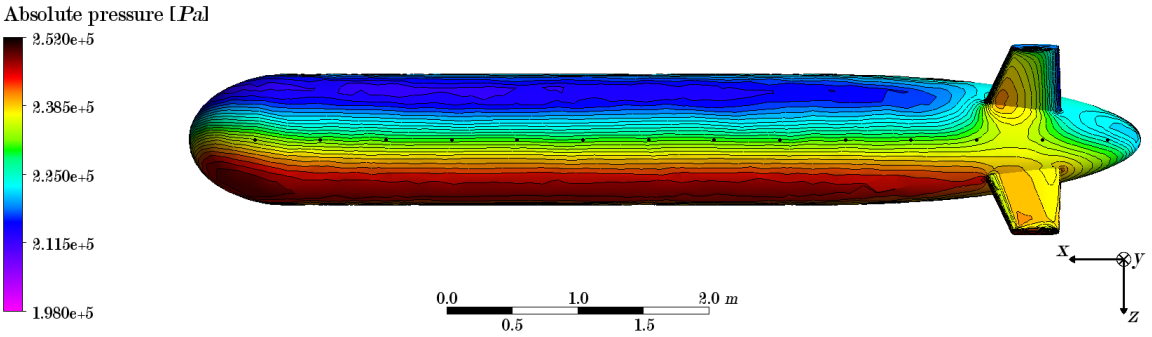
Figure 3.5: Upward motion at 5 kn.

In fig. 3.5 the absolute pressure distribution during an upward motion at 5 kn is shown. Again the starboard view is not given due to symmetry.

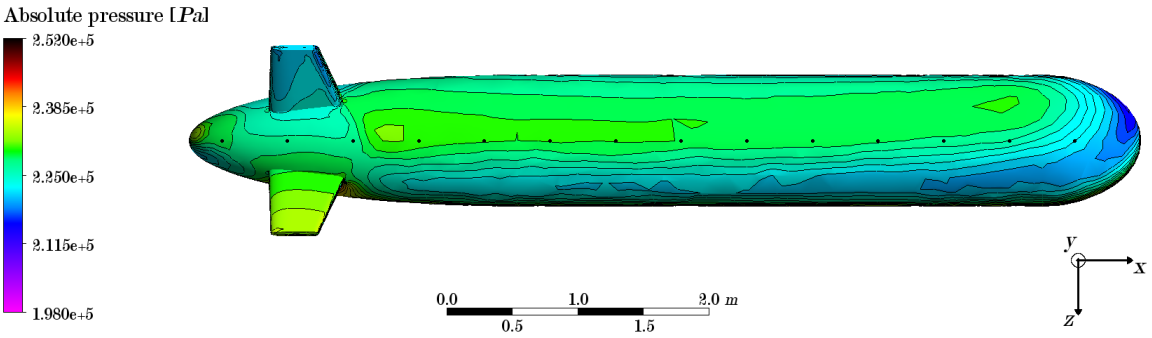
As can be seen in the top fig. 3.5(a) and the bottom fig. 3.5(c) view the pressure is slightly increased at the top and slightly reduced at the lower side of the AUV hull. However, the increase and reduction are not very distinct. But both can be caught by the upper and lower pressure measurement points.



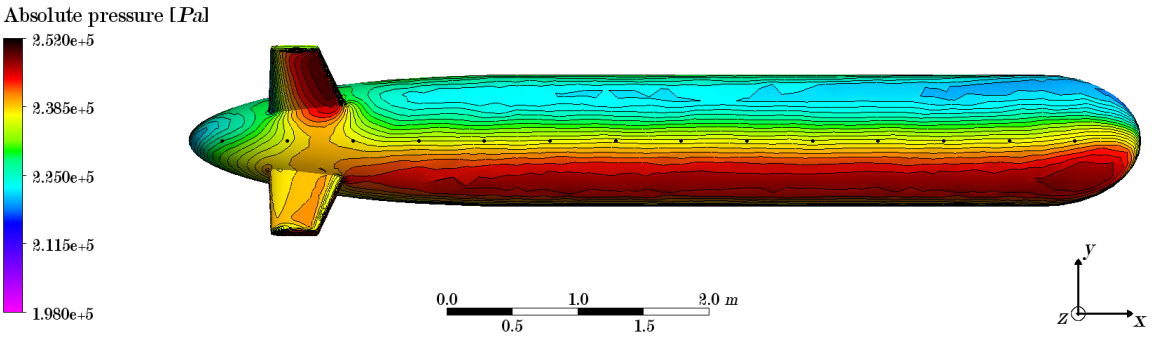
(a) Top view.



(b) Port view.



(c) Starboard view.



(d) Bottom view.

Figure 3.6: Combined motion of 5 kn forward, 5 kn downward, and 5 kn to port.

The port view fig. 3.5(b) again reveals strong turbulence between the fins. In addition the upper surfaces of the lower fins experience strong high pressures. This is not true for the upper surfaces of the upper fins (see fig. 3.5(a)). So the qualitative picture deviates compared to the downward and sideward motion results as in these cases the fins on the upwind side of the flow experience the highest pressure.

The slightly different behaviour of the pressure at the fins is due to a strong influence of the static pressure. As the pressure is already lowest at the topside the reduced velocity at the upper surfaces of the upper fins will result in a slightly higher but still low pressure. On the other hand the pressure is already higher on the upper sides of the lower fins. The additional pressure due to the flow plus the turbulence produced by the flow around the upper fins will result in an overall significantly higher pressure.

Fig. 3.6 above shows the resulting pressure distribution for a skew angle of attack of the water flow. The motion of the AUV is such that the components of the velocity are $5\,kn$ forward, $5\,kn$ downward, and $5\,kn$ to port. Mirroring the results at the z-x-plane will also yield the results for a motion with a velocity component of $5\,kn$ to starboard.

A region of high pressure can be observed at the lower port side extending from the nose back to the lower port side fin (see fig.'s 3.6(b) and (d)). This region is captured by the most forward lower and port side pressure measurement points. But most of the lower pressure measurement points are also near this region where the pressure is also significantly increased. Further regions of high pressure are also observed at the lower side of the upper port side fin, the upper side surface of the lower port side fin (fig. 3.6(b), not very distinct), and the lower side of the lower starboard fin (fig. 3.6(d)). The latter also yields the highest pressure over the whole surface.

A very distinct region of low pressure is situated at the upper port side extending from the upper portside fin to the nose (see fig.'s 3.6(a) and (b)). It also continues around the nose and the lower starboard side back to the lower starboard side fin (see fig.'s 3.6(c) and (d)). This region is well-captured by the upper pressure measurement points and especially the upper forward points. So the area away from the flow (upper starboard side) is not the region of lowest pressure.

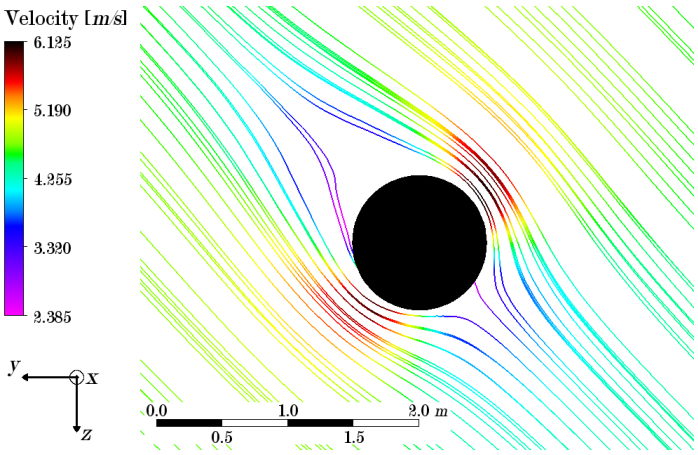
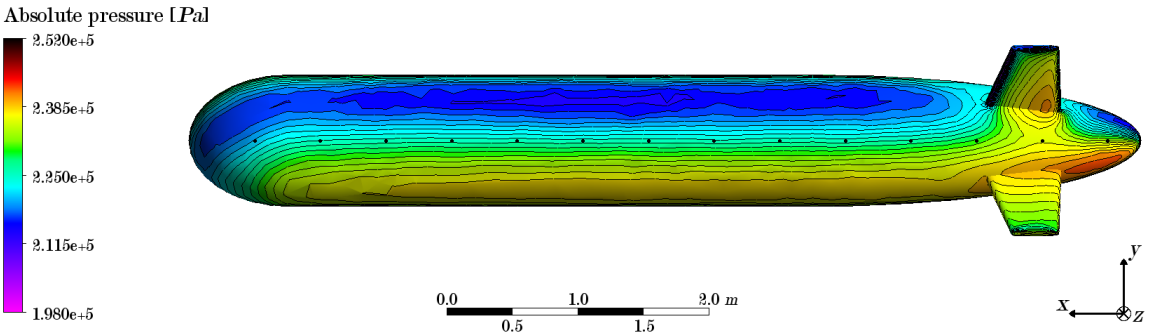


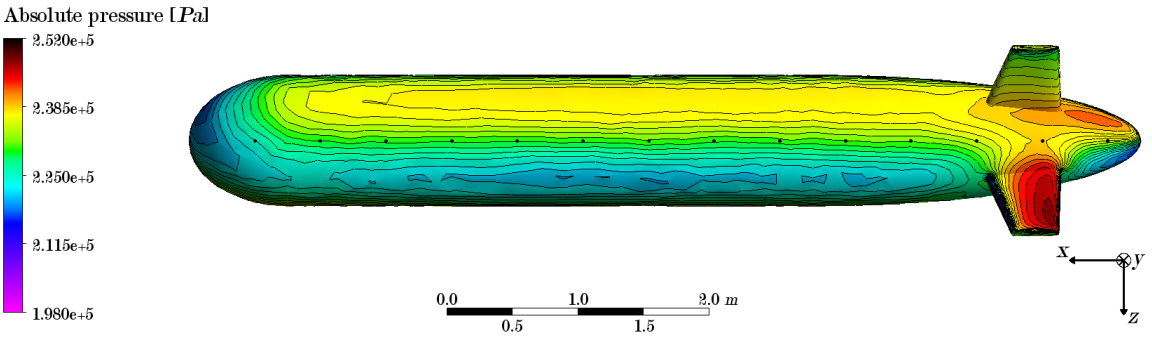
Figure 3.7: Front view of the AUV with velocity streamlines for the combined motion.

The reason for the observed pressure distribution can be seen in fig. 3.7. Following the streamlines around the AUV body it can be observed that the water flow is strongly decelerated comming toward the lower port side to about 50 % of the free flow velocity ($8.66\,kn \widehat{=} 4.46\,ms^{-1}$). Therefore,

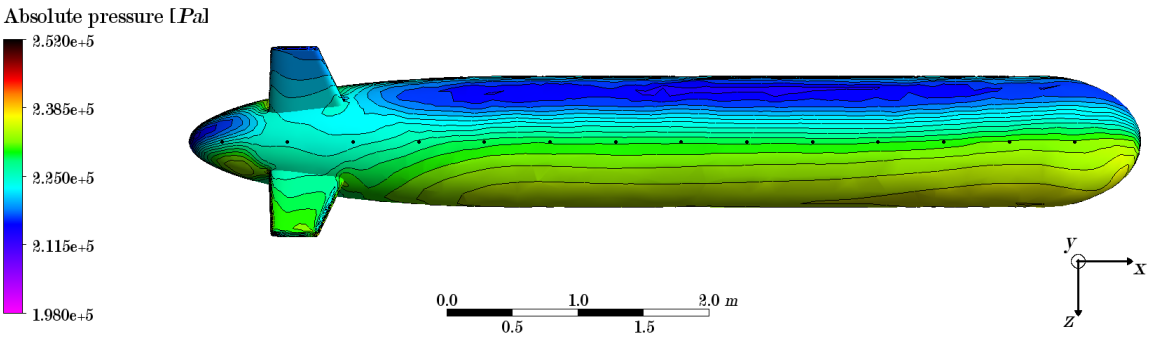
the pressure is strongly increased. When the flow goes around the cylindrical midsection of the hull the flow is accelerated to about 150 % of the free flow velocity at the upper port side and the lower starboard side. Hence, a significant decrease in pressure is present in this region.



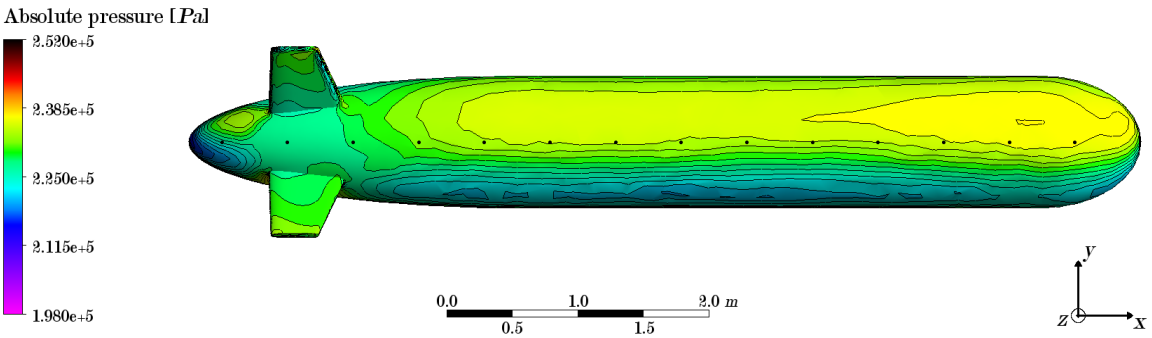
(a) Top view.



(b) Port view.



(c) Starboard view.



(d) Bottom view.

Figure 3.8: Combined motion of 5 kn backward, 5 kn upward, and 5 kn to port.

As can be expected the skew angle of attack yields a very diffferent picture of the pressure distribution compared to single-component flow velocities. The regions of different pressures are much more distinct as the absolute velocity is higher with the same velocity components as for

the flows along the coordinate axes. The pressure regions are also captured better by the pressure measurement points.

Fig. 3.8 above shows the resulting pressure distribution for a skew angle of attack of the water flow. The motion of the AUV is such that the components of the velocity are $5\,kn$ backward, $5\,kn$ upward, and $5\,kn$ to port. As before mirroring the results at the z-x-plane will also yield the results for a motion with a velocity component of $5\,kn$ to starboard.

The pressure distribution shows great qualitative similarities compared to the previous flow situation. Again one can observe a large high pressure region on the upwind side along the surface of the hull (upper port side, see fig.'s 3.8(a) and (b)) although it is not as distinct. This region is captured by the aft port side pressure measurement points. High pressures are also excerted on the upper surface of the lower port side fin which is also the highest pressure on the whole surface. Another high pressure region is found at the lower forward starboard side (fig. 3.8(d)). The lower forward pressure measurement points are within this region.

A low pressure region is situated along the upper starboard side (fig.'s 3.8(a) and (c)) continuing around the nose of the AUV and with lower intensity along the lower port side (fig.'s 3.8(b) and (d)). The sternmost upper and starboard pressure measurement points are in this region as well as some of the upper and port side forward points.

The observed pressure distribution is governed by the same principles as the previous flow situation (fig. 3.7). The water flow is strongly decelerated at the upper port side and then accelerated above the free flow velocity at the upper starboard side and the lower port side.

3.1.2 Forward / Backward Flow Velocity

Tab. 3.1 shows the root mean square errors (RMSE) in knots for all combinations of the 5 machine learning methods and the 5 different input data for the forward / backward velocity component. As it is only an RMSE the actual difference between the velocity obtained from the algorithms and the expected velocity can deviate significantly.

RMSE [kn]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	8.27	16.84	1.03	3.39	2.10
FFT	8.23	16.83	0.99	3.40	2.10
Conv	8.29	-	2.32	-	2.36
MA	7.83	-	5.06	6.66	2.32
SVD	9.06	-	4.64	6.10	5.41

Table 3.1: Root means square errors for the forward / backward flow velocity.

As one can see k-nearest neighbour (KNN) shows a large RMSE for raw data input (Raw) and fast Fourier transformed input (FFT) of above $16\,kn$. Furthermore, no meaningful results can be obtained for the other three input methods. For artificial neural networks (ANN) the RMSE is just above $8\,kn$ for all input methods except for moving averages (MA) and singular value decomposition (SVD) where the RSME is just below $8\,kn$ and just above $9\,kn$ respectively. In contrast these two input methods result in the highest RMSE for Bayesian networks (BN) of more than $6\,kn$. With convolution (Conv) no meaningful results are obtained in this case and for the other input methods the RMSE for Bayesian networks is below $3.5\,kn$. For multiple

linear regression (MLR) the root mean square error is below 2.5 kn except with singular value decomposition where it is more than twice as high. Support vector machines (SVM) yield the lowest root mean square error when using raw data or fast Fourier transformed data (around 1 kn). With the other input methods the root mean square errors are significantly higher.

Surprisingly, Bayesian networks perform much better than some of the other methods although it is expected that the pressure / velocity relation is not probabilistic. The good performance of multiple linear regression suggests that the relation is to some extend linear which is a sensible assumption.

In general, pre-processing the pressure data reduces the performance of the learning algorithms significantly. The exception is fast Fourier transformation which increases the performance slighlty in cases of ANN, KNN, and SVM. For BN the performance in slightly reduces with FFT.

Combined Method

With 5 learning machines and 5 input methods a potentially very large number of combinations (in the order of 10^{25}) are possible to increase the accuracy and output stability of the whole system. It is not feasible to test all of them. However, tab. 3.1 gives a good indication which methods are of interest for further study. These are SVM with raw, FFT, and convoluted input, and MLR with all input methods except SVD.

The best solution can be obtained using only support vector machines combined with raw, FFT, and convoluted input. The resulting root mean square error is 0.99 kn. This value is only slightly lower than using SVM with FFT input. However, the output is more stable, i.e. the variation is reduced.

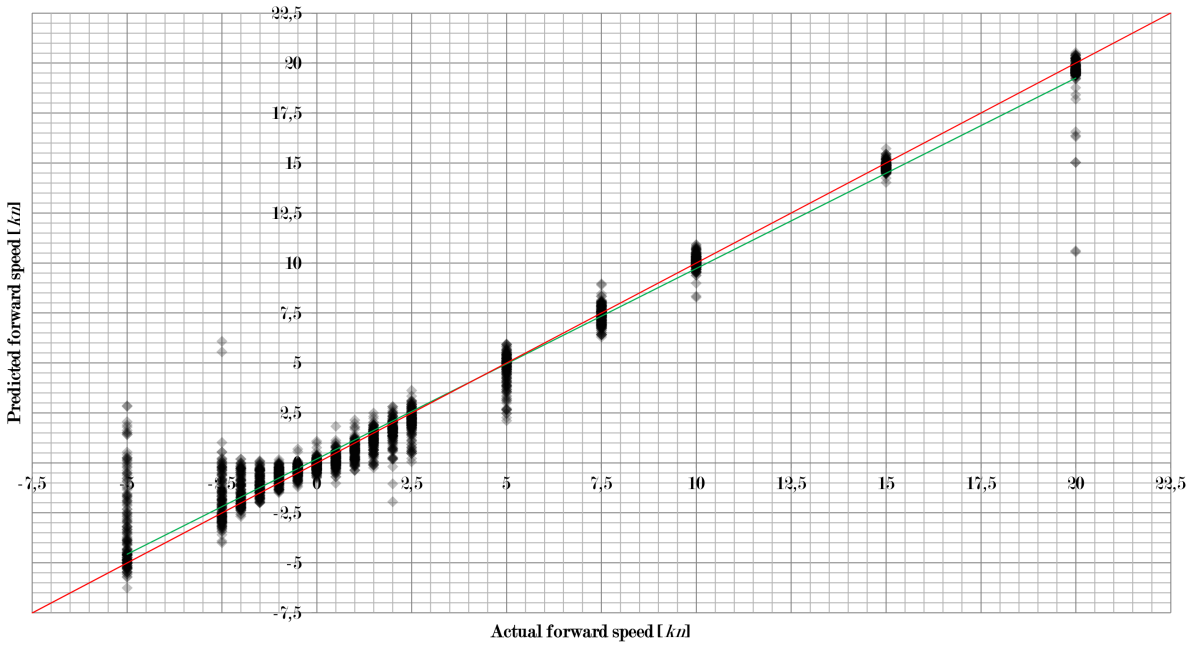


Figure 3.9: Predicted forward speed compared to actual forward speed (negative means backward).

Fig. 3.9 shows how this combination of learning machines and input methods works. The forward

/ backward component of the flow velocity predicted by the combined system is shown on the ordinate and plotted against the actual flow velocity set in the simulations. This is done for all 2872 flow situations and the intensity of the diamonds indicates the number of results. Hence, a black diamond shows that there are a large number of results falling in this area.

As one can see quite a large number of results are very close to the perfect prediction (red line). However, there are some results that deviate significantly. So there are problems with some unique flow situations that need to be addressed (see below). It can also be observed that the linear fit of the data (green line) is very close to the perfect prediction as well which indicates that the overall performance is very good.

Fig. 3.9 also shows that the absolute error first increases and then decreases with increasing flow velocity. The peak is at $5\,kn$. This results in the relative error decreasing with increasing velocity. Hence, low velocities are more difficult to work with than higher velocities. However, one can also observe that the backward flow situation poses significant diffculties. Fortunately, the backward flow situation is very rare in the application.

It can be observed that all outliers produced in the range of $\pm 5\,kn$ for the forward velocity component are due to small sideward and upward / downward velocity components of $\pm 1\,kn$ or less. Hence, not only the size of the velocity components is important but also the total flow speed. If at least one of the velocity components is sufficiently high the forward component can also be obtained quite accurately even if it is small in absolute value.

The matter is different for higher forward speeds of $7.5\,kn$ and above. Here, the outliers occur for very specific combinations of velocity components. Tab. 3.2 shows for which velocities the forward component is determined with the largest error.

u $[kn]$	v $[kn]$	w $[kn]$	**Error in** u	
			Absolute $[kn]$	**Relative** $[\%]$
7.5	-5.0	5.0	1.4	18.7
7.5	-1.0	0.5	1.1	14.5
7.5	-1.0	1.0	1.1	14.7
7.5	0.0	0.5	1.1	14.4
7.5	0.0	1.0	1.2	16.0
7.5	0.5	0.5	1.1	14.2
7.5	0.5	1.0	1.1	14.7
7.5	5.0	5.0	1.5	19.4
10.0	1.0	5.0	1.7	17.1
10.0	1.5	5.0	1.7	16.6
20.0	-2.5	5.0	5.0	24.9
20.0	-5.0	-5.0	3.6	18.1
20.0	-5.0	1.0	1.2	6.1
20.0	-5.0	5.0	9.4	47.2
20.0	2.5	-5.0	1.6	8.0
20.0	2.5	5.0	5.0	24.9
20.0	5.0	-5.0	3.7	18.4
20.0	5.0	2.0	1.8	9.0
20.0	5.0	2.5	3.4	17.1
20.0	5.0	5.0	9.4	47.0

Table 3.2: Velocities which show the highest error for the forward component ($u \geq 7.5\,kn$).

With regard to the velocity components no clear pattern can be observed. The simulation results give no indication for a reason of the outliers either. Hence, the training of the support vector machines must result in such a setup of the machines that large errors are produced for these unique flow situations.

3.1.3 Sideward Flow Velocity

Tab. 3.3 shows the root mean square errors (RMSE) in knots for all combinations of the 5 machine learning methods and the 5 different input data for the sideward velocity component. Again, as it is only an RMSE the actual difference between the velocity obtained from the algorithms and the expected velocity can deviate significantly.

RMSE [kn]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	2.28	-	0.62	-	1.06
FFT	2.28	-	0.63	-	1.06
Conv	2.37	-	1.32	-	1.20
MA	2.34	-	2.34	-	1.13
SVD	2.50	-	1.80	-	2.17

Table 3.3: Root means square errors for the sideward flow velocity.

The results are very different to those for the forward / backward component. As one can see no meaningful results can be obtained for k-nearest neighbour (KNN) and Bayesian networks (BN). For artificial neural networks (ANN) the RMSE is just below $2.5\,kn$ for all input methods. For multiple linear regression (MLR) the root mean square error just above $1\,kn$ except with singular value decomposition where it is above $2\,kn$. Support vector machines (SVM) yield the lowest RSME when using raw data or Fourier transformed data (around $0.6\,kn$). With the other input methods the root mean square errors are between $1\,kn$ and $2.5\,kn$.

As mentioned the overall behaviour of the machine learning algorithms deviates when compared to the forward / backward flow velocity calculations. The root mean square errors are lower and k-nearest neighbour and Bayesian networks do not work at all although BN yield satisfactory results for the forward / backward flow velocity.

For the remaining machine learning algorithms the picture is very similar to the one before. Again, multiple linear regression gives good results and best results are obtained with raw data and FFT data combined with support vector machines. However, in this case FFT increases the error with SVM.

Combined Method

In case of the sideward flow velocity all learning machine / input method combinations which yield meaningful results are of interest for the combined system. So ANN, SVM, and MLR with all input methods have to be taken into account to increase the accuracy and output stability of the whole system.

The best solution can be obtained using only support vector machines combined with all 5 input methods. The resulting root mean square error is $0.62\,kn$. This value is slightly higher than the

mininum RSME using SVM with raw data input. However, the output is more stable, i.e. the variation is reduced.

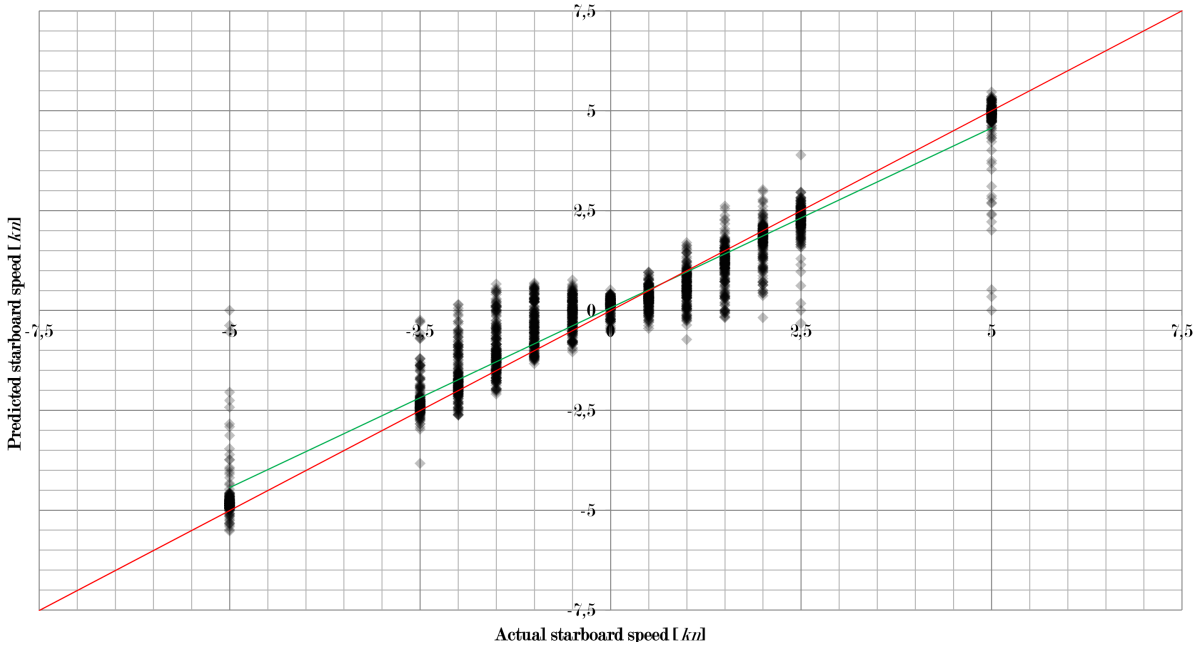


Figure 3.10: Predicted starboard speed compared to actual starboard speed (negative means port).

Fig. 3.10 shows how this combination of learning machines and input methods works. As before, the flow velocity predicted by the combined system is shown on the ordinate and plotted against the actual flow velocity set in the simulations. Again, this is done for all 2872 flow situations and the intensity of the diamonds indicates the number of results as before.

As one can see quite a large number of results are very close to the perfect prediction (red line). But there are also some significant deviations from the ideal line. It can also be observed that the linear fit of the data (green line) is very close to the perfect prediction as well. So the overall performance is also good in case of the sideward velocity component.

Fig. 3.10 also shows that the absolute error first increases and then decreases with increasing flow velocity as for the forward / backward component. The peak is at $2\,kn$. This results in the relative error decreasing with increasing velocity. A similar behaviour was already observed for the forward / backward velocity component. However, the largest errors occur for lower speeds in this case.

Having a closer look at all flow velocity components one may observe that large errors in the determination of the sideward component occur for different sizes of the forward / backward component. Hence, the forward / backward has no influence on the accuracy of the sideward component.

However, a strong influence of the upward / downward can be seen. Those points in fig. 3.10 which deviate strongly from the ideal line coincide with a large upward / downward component (mostly $5\,kn$ in absolute value). One has to consider how water flows around the AUV body for these flow situations and where the pressure measurement points are located in order to understand why this is the case.

Viewed from the front there is a strong qualitative similarity between these flow situations and the one observed in fig. 3.7. Since the pressure measurement points are positioned at the top, bottom, and the two sides of the AUV, flows with only a sideward or an upward / downward component can be captured very well as the position of the velocity head with its significant pressure increase can be determined quite easily. However, for a skew angle of attack the velocity head is in-between the measurement points and only medium pressure changes are observed by them. Therefore, it is difficult for the learning machines to distinguish between the influences of the flow components and to separate them.

3.1.4 Upward / Downward Flow Velocity

Tab. 3.4 shows the root mean square errors (RMSE) in knots for all combinations of the 5 machine learning methods and the 5 different input data for the upward / downward velocity component.

RMSE [kn]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	2.30	-	0.40	-	0.86
FFT	2.30	-	0.43	-	0.86
Conv	2.36	-	1.20	-	1.19
MA	2.50	-	2.25	-	0.98
SVD	2.83	-	2.68	-	2.44

Table 3.4: Root means square errors for the upward / downward flow velocity.

As one can see, again, no meaningful results can be obtained for k-nearest neighbour (KNN) and Bayesian networks (BN). For artificial neural networks (ANN) the RMSE is just below $2.5\,kn$ for all input methods except for singular value decomposition (SVD) where the RMSE slightly above this value. For multiple linear regression (MLR) the root mean square error is around $1\,kn$ except with singular value decomposition where it is above $2\,kn$. Support vector machines (SVM) again yield the lowest RSME when using raw data or Fourier transformed data (around $0.4\,kn$). With the other input methods the root mean square errors are significantly higher.

The overall picture is very similar to sideward flow velocity situation. Again multiple linear regression gives good results and best results are obtained with raw data and FFT data. However, the performance gap between raw and FFT input compared to the other input methods is larger. While raw and FFT inputs yield even smaller root mean square errors compared to the sideward flow velocity calculations, the performance of the other input methods is lower.

Combined Method

In case of the upward / downward flow velocity again all learning machine / input method combinations which yield meaningful results are of interest for the combined system as for the sideward flow velocity component. So ANN, SVM, and MLR with all input methods have to be taken into account.

The best solution can be obtained using only support vector machines combined with all 5 input methods. The resulting root mean square error is $0.40\,kn$. This value is slightly higher than the mininum RSME using SVM with raw data input. However, the output is more stable, i.e. the variation is reduced.

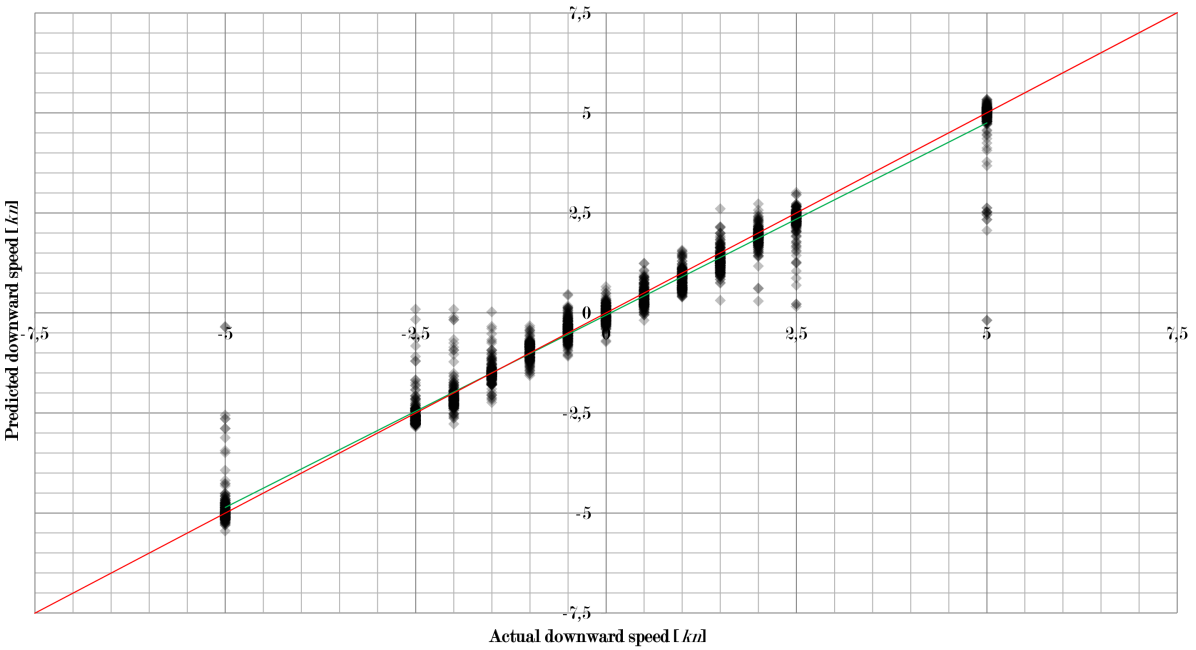


Figure 3.11: Predicted downward speed compared to actual downward speed (negative means upward).

Fig. 3.11 shows how the combination of learning machines and input methods works. Again, the is plotted against the actual flow velocity set in the simulations.

As one can see quite a large number of results are very close to the ideal line which could also be seen for the other two velocity components. However, some significant deviations in the results are observed again. It can also be seen that the linear fit of the data (green line) is very close to the perfect prediction as well which indicating a very good performance.

Fig. 3.11 also shows that the absolute error first increases and then decreases with increasing flow velocity as for the other flow velocity components. The peak is at $1.5\,kn$ and therefore even lower than for the sideward component. This results in the relative error decreasing with increasing velocity. Hence, low velocities are more difficult to work with than higher velocities. However, higher velocities also produce more outliers. But it can also be observed that the number of outliers is much lower when compared to the other velocity components.

In addition, one may observe that fig.'s 3.10 and 3.11 are very similar. This is sensible as the influence of the upward / downward velocity component on the accuracy of the sideward component also acts in the opposite direction. Those points in fig. 3.11 which deviate strongly from the ideal line coincide with a large sideward component (mostly $5\,kn$ in absolute value). The reason for the mutual influence was already discussed in section 3.1.3.

As for the sideward component large errors in the determination of the upward / downward component occur for different sizes of the forward / backward component. Hence, the forward / backward has no influence on the accuracy of the upward / downward component.

3.2 Underwater Image Processing

3.2.1 Red Channel

Tab. 3.5 shows the root mean square errors (RMSE) for all combinations of the 5 machine learning methods and the 5 different input data for the red channel of 24 bit RGB images, i.e. every channel has 256 possible pixel values. As mentioned for the flow velocity the actual deviation of a pixel colour from the expected value can deviate significantly.

RMSE [px]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	65.71	37.95	12.60	77.31	73.90
FFT	65.43	7.17	13.04	77.31	73.90
Conv	65.59	10.22	13.19	77.31	73.90
MA	74.32	67.40	69.52	79.26	73.92
SVD	65.86	37.95	12.60	77.31	73.90

Table 3.5: Root means square errors for the red channel.

As one can see Bayesian networks (BN) and multiple linear regression (MLR) show a large RMSE independent of the input data of well above $70\,px$. Also artificial neural networks (ANN) show a significant RMSE of around $65\,px$ for all types of input data except with moving averages where the RSME is much higher. For k-nearest neighbour (KNN) and support vector machines (SVM) the performance strongly depends on the input data. Both show a very large RMSE when the image data was smoothed with moving averages just below $70\,px$. KNN with raw data or data decomposed with singular value decomposition shows a medium RMSE around $38\,px$. Much lower RMSE are observed for SVM using any type of input data (RMSE about $13\,px$) except data smoothed with moving averages (RMSE above $67\,px$). The lowest root mean square errors are obtained for KNN combined with convoluted data (about $10\,px$) and FFT (about $7\,px$).

The low performance of Bayesian networks can be expected as image data is not probabilistic. Also the high RMSE for multiple linear regression is to be anticipated. It was already stated in section 1.4 that the assumption of a linear dependency between the colours as seen underwater and under sunlight does not hold beyond small depths. This is especially true for the red channel since the red colour is most strongly filtered in seawater (fig. 1.7) and the light source in the laboratory setup emits almost no light in the red region (fig. 2.12).

Smoothing the input data with moving averages (MA) reduces the performance of the learning machines. Especially, for KNN and SVM where the RSME is quite low the drop in performance is significant. It can therefore be deduced that small changes of the pixel values have a strong influence on the result. These small changes are removed by moving averages and hence the input data no longer represents the original image properly.

Combined Method

Again, as for the flow velocity analysis it is not feasible to test all possible learning machine and input method combinations for the increase of accuracy and output stability. Tab. 3.5 indicates that KNN with FFT and convoluted data and SVM with all input methods except moving averages are of interest for further study of the red channel.

The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input although k-nearest neighbour initially yields the lowest errors. The resulting root mean square error is $5.27\,px$. This improves the best result obtained with a single machine (KNN with FFT) by almost $2\,px$.

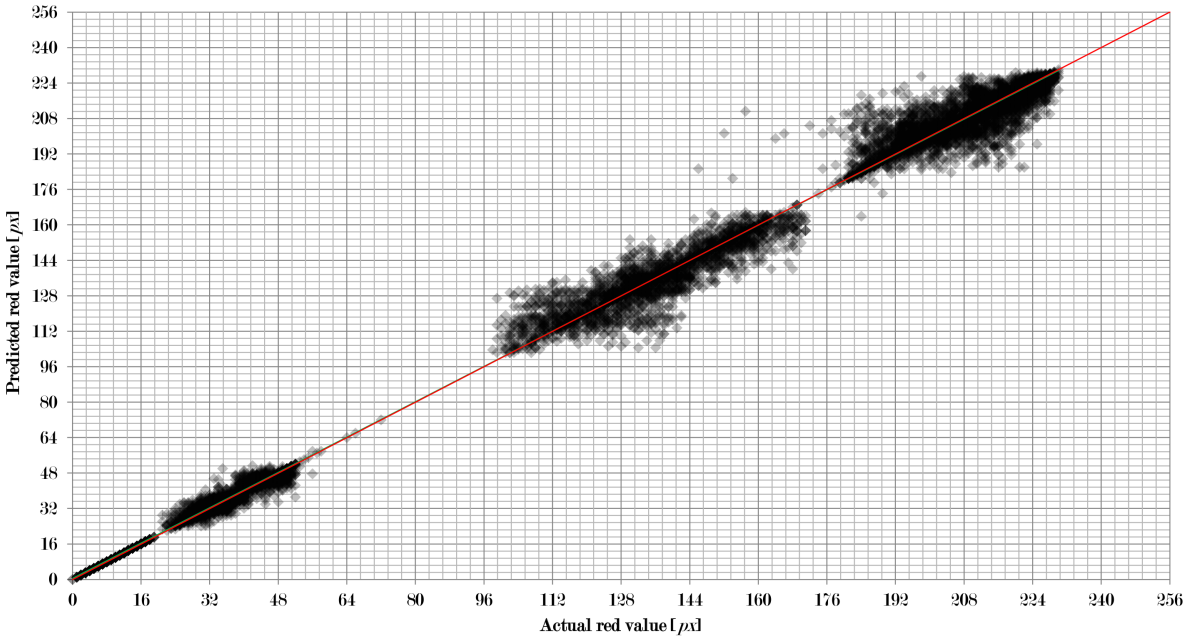


Figure 3.12: Predicted red value compared to actual red value.

Fig. 3.12 shows how this combination of learning machines and input methods works. The red value of a pixel predicted by the combined system is shown on the ordinate and plotted against the actual red value of the pixel obtained from an image under white light. This is done for a large sample of pixels (about 15500) taken from different test images. As before, the intensity of the diamonds indicates the number of results. Hence, a black diamond shows that there are a large number of results falling in this area.

As one can see quite a large number of results are very close to the perfect prediction (red line). There are some results that deviate. But the deviation is not very large. Only some single outliers can be seen that need to be addressed. It can also be observed that the linear fit of the data (green line) is almost on top of the line of perfect prediction indicates that the overall performance is excellent.

Fig. 3.12 also shows that the error generally increases with increasing red value. The is very sensible as it was already indicated that under underwater lighting conditions red colour is very difficult to capture and shows great similarities to dark grey and black.

3.2.2 Green Channel

Tab. 3.6 shows the root mean square errors (RMSE) for all combinations of the 5 machine learning methods and the 5 different input data for the green channel of 24 bit RGB images.

As one can see artificial neural networks (ANN) show a large RMSE independent of the input data of above $60\,px$. Also multiple linear regression (MLR) and Bayesian networks (BN) show a significant RMSE of around $40\,px$ to $45\,px$ for all types of input data although they are much

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	61.94	23.40	10.57	43.50	40.01
FFT	61.49	10.58	10.38	43.49	40.01
Conv	61.78	10.76	10.56	43.50	40.02
MA	66.27	75.12	53.62	45.05	40.02
SVD	61.85	23.40	10.67	43.50	40.01

Table 3.6: Root means square errors for the green channel.

lower than for ANN. For k-nearest neighbour (KNN) and support vector machines (SVM) the performance again depends on the input data. Both show a very large RMSE when the image data was smoothed with moving averages as for the red channel. In fact KNN with moving averages has the largest error (about $75\,px$) over all combinations. KNN with raw data or data decomposed with singular value decomposition shows a medium RMSE around $30\,px$. Much lower RMSEs (below $11\,px$) are observed for SVM using any type of input data except data smoothed with moving averages (RMSE above $53\,px$) and for KNN combined with convoluted data and FFT (RMSE also below $11\,px$).

The overall picture is very different to the results for the red channel. The performance of ANN, KNN, and SVM is slightly better. However, the root mean square error for BN and MLR are is reduced by 45 % compared to the error for the red channel. The better performance of multiple linear regression is due to the fact that green colour is not as strongly filtered as red colour in seawater (fig. 1.7) and the light source in the laboratory setup emits a significant amount of light in the green region (fig. 2.12). Again smoothing the input data with moving averages (MA) reduces the performance of the learning machines which confirmes the assumption that this removes important data in the input.

Combined Method

In case of the green channel KNN with FFT and convoluted data and SVM with all input methods except moving averages are of interest for the combined system. The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. Again k-nearest neighbour drops out although the single methods give very good results. The resulting root mean square error is $3.95\,px$. This improves the best result obtained with a single machine (SVM with FFT) significantly as the RMSE is now less than 40 % of the lowest single method error.

Fig. 3.13 shows how this combination of learning machines and input methods works. The green value of a pixel predicted by the combined system is shown on the ordinate and plotted against the actual green value of the pixel obtained from an image under white light. As before, the intensity of the diamonds indicates the number of results.

As one can see quite a large number of results are very close to the perfect prediction (red line). However, there are some results that deviate significantly. It can also be observed that the linear fit of the data (green line) is even closer to the ideal line than for the red channel. So theoverall performance is also very good.

Fig. 3.13 also shows that the error is slightly lower for the middle range than for larger and smaller

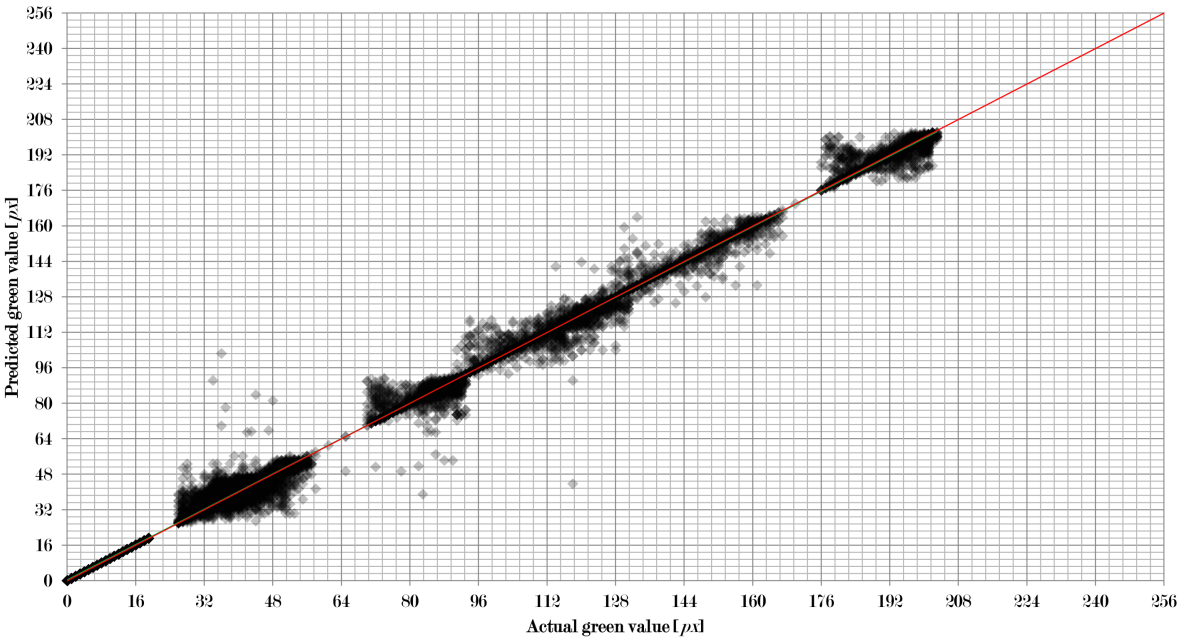


Figure 3.13: Predicted green value compared to actual green value.

pixel values. But the difference is very small. At the same time more outliers are observed in this range.

3.2.3 Blue Channel

Tab. 3.7 shows the root mean square errors (RMSE) for all combinations of the 5 machine learning methods and the 5 different input data for the blue channel of 24 bit RGB images.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	43.64	30.03	10.79	-	17.77
FFT	43.60	10.21	10.74	-	17.77
Conv	43.57	11.12	11.54	-	17.78
MA	46.30	-	21.24	-	17.78
SVD	43.70	30.03	10.79	-	17.77

Table 3.7: Root means square errors for the blue channel.

As one can see no meaningful results could be obtained for Bayesian networks (BN). The same is true for k-nearest neighbour (KNN) when the input data is smoothed with moving averages (MA). Artificial neural networks (ANN) show a significant RMSE of around $45\,px$ for all types of input data. With a value below $18\,px$ the RSME is much lower for multiple linear regression (MLR) for all imput methods. As before, the performance of k-nearest neighbour (KNN) and support vector machines (SVM) depends on the input data. KNN with raw data or data decomposed with singular value decomposition shows a medium RMSE around $30\,px$. Much lower RMSE are observed for SVM using any type of input data except data smoothed with moving averages (RMSE above $21\,px$) and for KNN combined with convoluted data and FFT (RMSE between $10\,px$ and $12\,px$).

The results are again very different when compared to the red and the green channel. The performance of ANN and MLR are significantly better when compared to the green channel. The

good performance of multiple linear regression makes sense as blue colour is least strongly filtered in seawater (fig. 1.7) and the light source in the laboratory setup emits light mostly in the blue region of the spectrum (fig. 2.12). Again smoothing the input data with moving averages (MA) reduces the performance of the learning machines.

Combined Method

In case of the blue channel again KNN with FFT and convoluted data and SVM with all input methods except moving averages are of interest for the combined system. In addition multiple linear regression also shows promise in this case as the RMSE is not very high for all input methods.

The best solution can be obtained using only support vector machines combined with all input methods. Hence, again k-nearest neighbour drops out and multiple linear regression is not considered as well. The resulting root mean square error is $2.85\,px$. This improves the best result obtained with a single machine (KNN with FFT) significantly as the RMSE is reduced by more than 70 % compared to the lowest single method error.

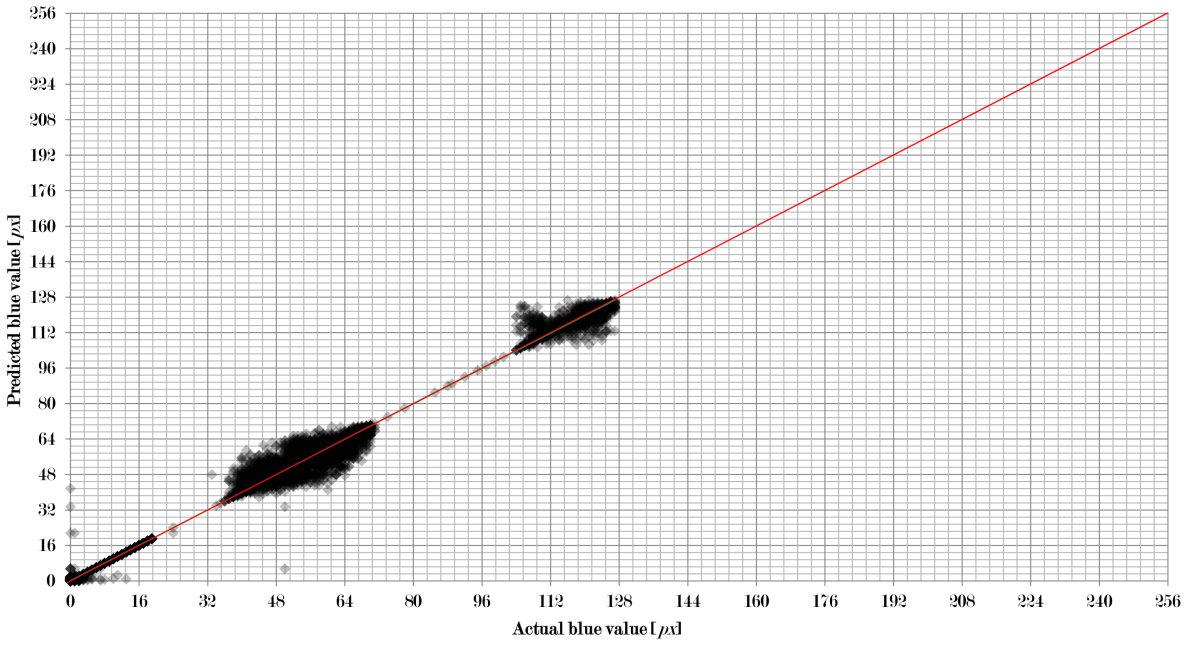


Figure 3.14: Predicted blue value compared to actual blue value.

Fig. 3.14 shows how this combination of learning machines and input methods works. The blue value of a pixel predicted is shown plotted against the actual blue value of the pixel obtained from an image under white light as before.

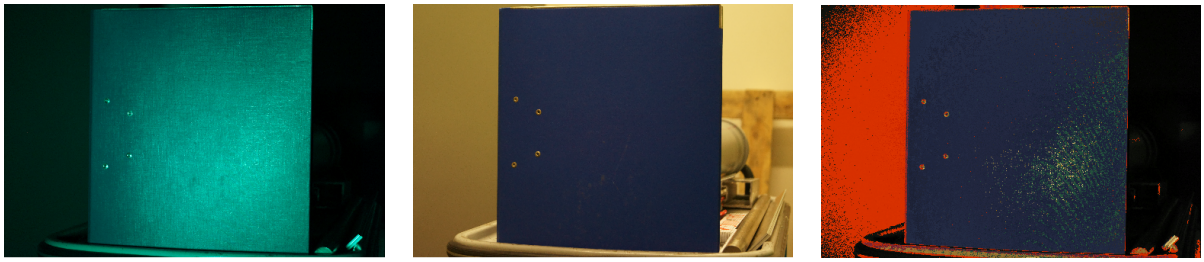
As one can see quite a large number of results are very close to the perfect prediction (red line). Some deviations can be observed as well. But in general the results are very good. The linear fit of the data (green line) is now in line with the ideal line indicating that the overall performance is excellent.

Fig. 3.14 also shows that the error generally decreases with increasing blue value. The is sensible as blue colour is very well captured in case of the underwater lighting condition.

3.2.4 Image Results

Fig. 3.15 shows how the approach in this thesis works for a blue sample object in the laboratory setup. The left image fig. 3.15(a) shows the object illuminated by the special light source (underwater lighting conditions), the middle image fig. 3.15(b) shows the same object under white light, and the right image fig. 3.15(c) shows the reconstruction of the object done by the combined machine learning algorithm using the underwater lighting image fig. 3.15(a).

As one can see the colours of the object under white light are reproduced quite well. Problems occur at the lower right of the the object where some green and white pixels are produced instead of blue ones. Also the background is not well-reproduced.

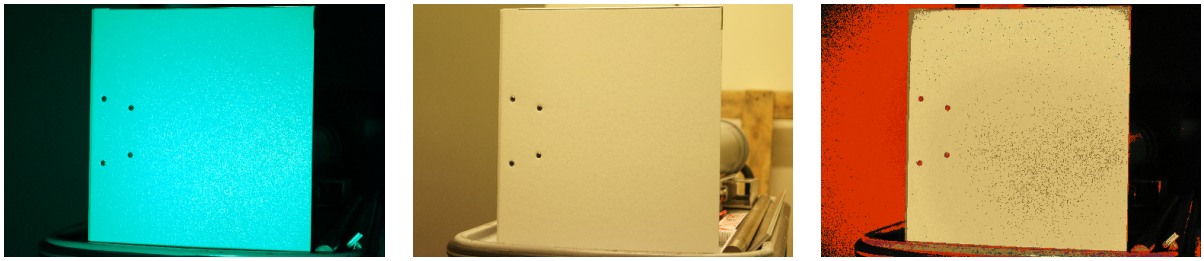


(a) Underwater lighting. (b) White light. (c) Reconstruction.

Figure 3.15: Results for a blue object.

Due to the strong and focused light source as well as the smooth surface of the sample objects reflections occur in some regions of the image under underwater lighting conditions. The scenery is not uniformly lighted as for the object under white light. In these parts the colour reconstruction is not as good as in the rest of the image. Reflections change the relation between the colours under white light and under underwater lighting conditions. This results in the green and white patches on the object.

The background in image fig. 3.15(a) is completely black and the objects in the background are not visible at all. Hence, one cannot expect theses objects to be reconstructed. However, the colour of the background should also be black in the reconstructed image. This is not the case as part of the background is red in image fig. 3.15(c). The reason is that red light is strongly influenced by the lighting conditions (see fig. 1.7). It can be observed that red objects look almost the same as black and dark grey objects (see also the discussion on the red object below). Hence, regions which appear in these colours in the underwater image have to be mapped both to red and to black or dark grey. It is clear, that this does not work in every situation.

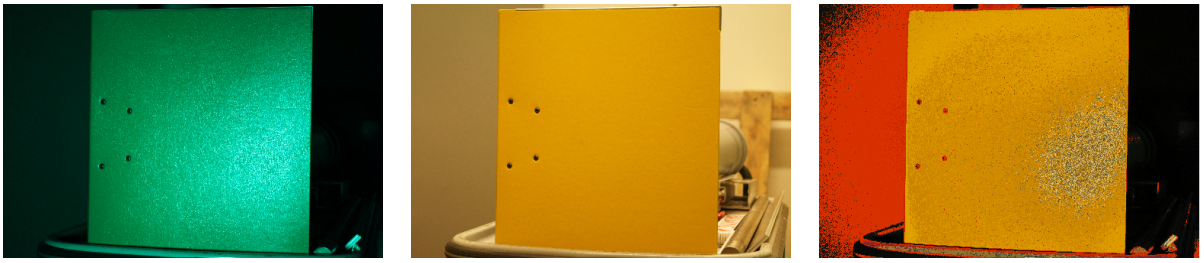


(a) Underwater lighting. (b) White light. (c) Reconstruction.

Figure 3.16: Results for a white object.

In fig. 3.16 one can see the results for a white sample object. Again the reconstruction is quite well with some slightly darker pixels destributed in the lower left part of the object. Again the background is not well-reproduced. Some brownish patches can also be observed at the boundary between the object and the background.

Again reflections pose a problem for the colour reconstruction. With the white sample object these can be clearly observed for the underwater lighting conditions (fig. 3.16(a)) although not as strongly as for the blue object. The brownish colour at the object boundaries is due to the fact that the boundaries are not well captured in fig. 3.16(a).



(a) Underwater lighting. (b) White light. (c) Reconstruction.

Figure 3.17: Results for a yellow object.

In the above fig. 3.17 the results for the regonstruction of a yellow sample object are presented. The reconstruction is not as good as for the previous two sample objects. There is a patch of strong colour deviation to the right and the colour reconstruction for the remaining object also shows some differences. Again the background is not well-reproduced as well.

When looking at fig. 3.17(b) the sample objects seems to have a smooth surface. However, fig. 3.17(a) reveals that this is not the case. Also the surface is very reflective. Combined with the focused light source this results in a very inhomogenous illumination of the surface under underwater lighting condition. Hence, the quality of the colour reconstruction varies strongly over the whole surface.

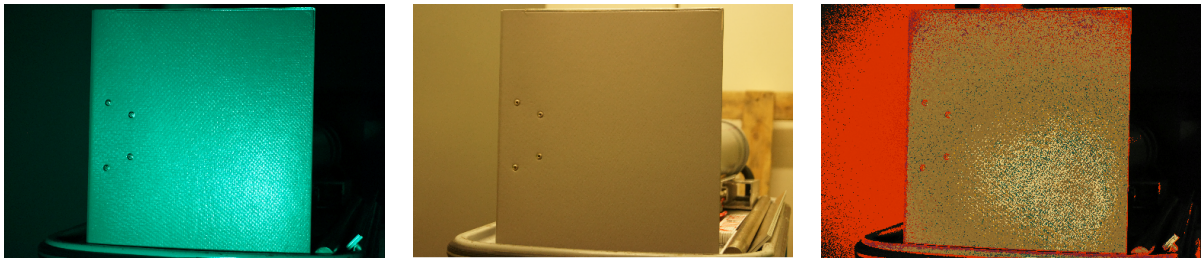
Fig. 3.18 below shows the machine learning results for a green sample object. The deviations between the white lighting condition (fig. 3.18(b)) and the reconstruction (3.18(c)) is even greater than for the yellow sample objects. In the lower right of the object a large white patch is produced. For the remaining object surface a significant amount of blue pixels is produced instead of green. Furthermore, some red can be observed at the boundaries at the top and especially at the upper right corner.

(a) Underwater lighting. (b) White light. (c) Reconstruction.

Figure 3.18: Results for a green object.

Looking at fig.'s 3.18(a) and (b) one can observe that the upper right corner is slightly bent backwards. Hence, there is a shadow and the colour values for the underwater lighting conditions are very similar to that of a blue object as in fig. 3.15(a). Also the object surface is quite rough and reflective as for the yellow object before. Therefore, one can see the same problems due to reflection with the reconstruction of the colour for the green object.

Fig. 3.19 shows how the approach works for a grey sample object. For the underwater lighting conditions (a) one can observe structuring of the surface especially in the lower right part of the object. However, the structure is not as pronounced as for the yellow and green sample objects. Also the surface is not as reflective. The upper edge and the two upper corners are not illuminated strongly and are slightly bent backwards such that this part appears to be darker than the rest of the object.



(a) Underwater lighting. (b) White light. (c) Reconstruction.

Figure 3.19: Results for a grey object.

In the reconstructed image (fig. 3.19(c)) the grey colour of the object is well represented. However, there are a number of green pixels distributed over the whole surface. In addition, a significant amount of red pixels can be found at the upper edge and the upper corners. The strongly illuminated lower right of the object is reconstructed with much brighter colour than expected.

Under underwater lighting conditions both the grey object (fig. 3.19(a)) and the green object look (fig. 3.18(a)) very similar. Hence the pixel values are also very close together which results in some pixels obtaining a green instead of a grey colour through the algorithm. The bright colours in the lower right are again due to the strong focused light source which changes the relation between the colours under white light and under underwater lighting conditions.

The following figure (fig. 3.20) one can see the results for a purple sample object. The scenery under underwater lighting conditions (a) is quite dark with a brighter region in the lower middle of the object. Also little reflection can be observed. The reconstruction (c) mainly shows purple colour with red and dark purple distributed over the whole object. In the lower middle some green and brownish pixels can be found as well.

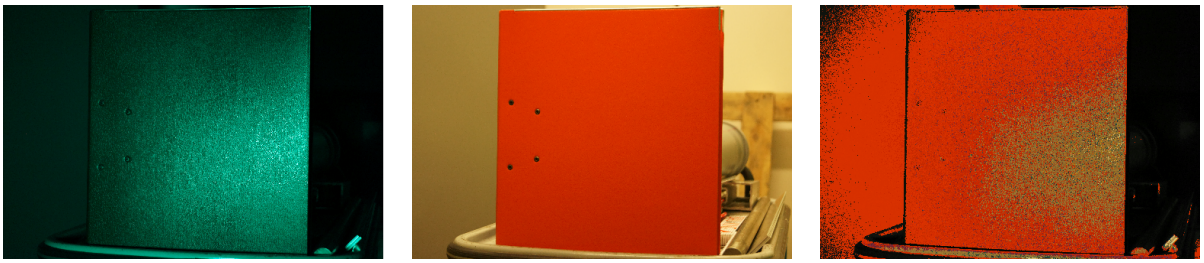
The colour reconstruction is quite good in general. Comparing fig. 3.20(a) to fig. 3.15(a) one may observe some similarities in the colour especially to the left. Hence, one can expect some blue pixels in the resulting image. The red pixels are due to the fact that purple contains both red and blue. The red channel is reconstructed correctly but the blue channel is not. Instead the dark regions are mapped to black / dark grey in case of the blue channel. The combination therefore gives red colour. The green and brownish pixels in the lower middle are in the slightly brighter region of the object observed in fig. 3.20(a). So these are due to a different relation

(a) Underwater lighting. (b) White light. (c) Reconstruction.

Figure 3.20: Results for a purple object.

between the colours under underwater lighting condition and under white light. Reflection does not cause any problems in this case.

Fig. 3.21 below shows the machine learning results for a red sample object. Again the scenery under underwater lighting conditions (a) is very dark with significant reflections to the lower right of the object. The reconstruction (c) produces mostly red pixels for the object with some distributed dark purple pixels. To the lower right there is a significant amount of brownish pixels and some green pixels as well.



(a) Underwater lighting. (b) White light. (c) Reconstruction.

Figure 3.21: Results for a red object.

The red colour is reconstructed very well and most pixels in the object are of this colour. The dark purple pixels are due to a similar problem as for the previous object in purple. Again the red channel is well reconstructed and the blue channel is not. But in this case there is too much blue in the reconstruction such that one obtains purple instead of red. The green and brownish pixels in the lower right are again due to significant reflections as in many of the other objects.

4 Improved Underwater Navigation

4.1 Adapted Position of Pressure Measurement Points

4.1.1 Motivation

In section 3.1.1 it can be observed that the changes in pressure near the pressure measurement points are not the most significant ones over the whole surface of the AUV body. Other regions are influenced by the changes in velocity much stronger. It is sensible to assume that by moving the pressure measurement points to regions with higher pressure variations the performance of the learning machines can be increased.

Furthermore, an RMSE of up to $1\,kn$ (for the forward / backward component) is too large for navigational purposes. Also the number of pressure measurement points is quite high. A more optimal positioning may allow a smaller number of measurement points which in turn reduces the complexity of the learning machines.

At the moment up to five learning machines are applied in parallel. This requires a considerable amount of computational power. It is desirable to use only one or two learning machines instead. An increased performance of the individual machines due to a better position of the pressure measurement points may also reduce the number of learning machines required to get a sufficiently accurate output.

4.1.2 Adjustment

Looking at fig.'s 3.1 to 3.8 one can make the following observations:

1. Sideward, upward, and downward flow velocity components result in significant pressure changes on the fins. There are distinct high pressure regions on the upstream side and low pressures on the downstream side.
2. Forward and backward flow velocity components result an increase in pressure near the bow and the stern respectively.
3. The isobars on the AUV body are parallel and almost horizontal in all flow situations.
4. For skew angles of attack the pressure changes are not well-captured, as the most significant changes and the velocity head are in between the lines on which the pressure measurement points are distributed.

These observations lead to the following changes in the positioning of the pressure measurement points:

1. Place measurement points on both sides of each fin to capture the significant pressure changes and especially the differences between the upstream and the downstream sides.
2. Keep the most forward and most backward pressure measurement points as these are the only points which can capture the changes due to a forward and backward flow velocity component.

3. It is sufficient to have one set of measurement points on the cylindrical part of the AUV body as the points along one line measure almost the same pressure and are therefore redundant.

4. Introduce additional pressure measurement points in between the lines where the measurement points are currently placed such that the points are not only at the top, the bottom, and the two sides.

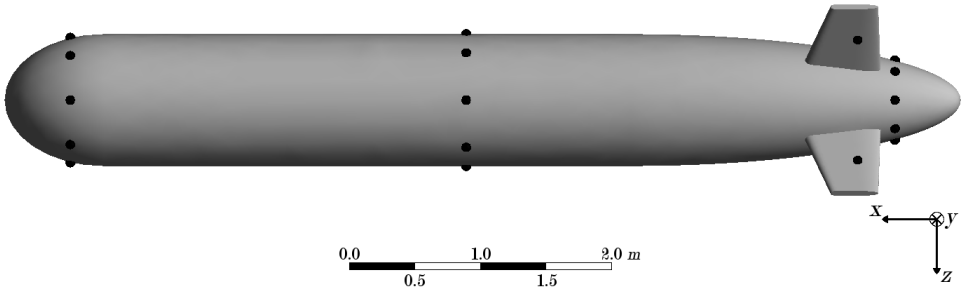


Figure 4.1: Position of pressure measurement points in the new setup.

Fig. 4.1 shows the resulting positions of the measurement points when the above steps are applied. As one can see there are now 4 sets of 8 points each which gives a total of 32 pressure measurement points. This is much less than the original 56 points. Hence, a reduction of the number of measurement points is also achieved. This will also give less complex learning machines as mentioned before.

4.2 Results and Discussion

4.2.1 Forward / Backward Flow Velocity

Tab. 4.1 shows the root mean square errors (RMSE) in knots for all combinations of the 5 machine learning methods and the 5 different input data for the forward / backward velocity component with the reduced number and positionwise adjusted pressure measurement points. As it is only an RMSE the actual difference between the velocity obtained from the algorithm and the expected velocity can deviate significantly.

RMSE [kn]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	8.02	19.07	1.20	3.77	2.65
FFT	7.97	18.97	1.13	3.83	2.65
Conv	8.18	-	2.74	-	2.84
MA	8.37	-	4.38	4.02	2.77
SVD	8.29	-	2.25	3.07	3.06

Table 4.1: Root means square errors for the forward / backward flow velocity.

As one can see k-nearest neighbour (KNN) shows a large RMSE for raw data input (Raw) and fast Fourier transformed input (FFT) of above $18\,kn$. Furthermore, no meaningful results can be obtained for the other three input methods. For artificial neural networks (ANN) the RMSE is just above $8\,kn$ for all input methods except for FFT where the RSME is just below $8\,kn$. Singular value decomposition (SVD) results in the lowest RMSE for Bayesian networks (BN) of

just above $3\,kn$. With convolution (Conv) no meaningful results are obtained in this case and for the other input methods the RMSE for Bayesian networks is around $4\,kn$. For multiple linear regression (MLR) the root mean square error is a little bit over $2.5\,kn$ except with singular value decomposition where it is more than $3\,kn$. Support vector machines (SVM) yield the lowest root mean square error when using raw data or fast Fourier transformed data (just above $1\,kn$). With the other input methods the root mean square errors are between $2\,kn$ and $4.5\,kn$.

When compared to the original results (tab. 3.1) the overall performance is slightly better. However, some of the input data / learning machine combinations show a significant decrease in performance while others perform very well. Particularly, all learning machines perform much better with SVD data input compared to the original setup. Artificial neural networks do not show very much change. There are some slight increases and decreases in the RMSE depending on the input data. Support vector machines, Bayesian networks, and multiple linear regression all show a small decrease in performance with the exception of SVM and BN with moving averages where the error is reduced. Finally, k-nearest neighbour which already shows bad results in the original setting is performing even worse with the new setup.

Multiple linear regression shows good results although the RMSE is slightly higher that before. This suggests that the pressure / velocity relation in the new setup is again quite linear. As before, pre-processing the pressure data reduces the performance of the learning algorithms. The exception is again fast Fourier transformation which increases the performance slighlty in cases of ANN, KNN, and SVM as for the original setup.

Combined Method

In case of the forward / backward flow velocity SVM with raw, FFT, convoluted, and SVD input, BN with SVD input, and MLR with all input methods are of interest for the combined system. The other input data / machine learning combinations do not show sufficient performance.

The best solution can be obtained using only support vector machines combined with raw and FFT input. The resulting root mean square error is $1.13\,kn$. This value is the same as using SVM with FFT input. It is also slightly higher than for the original pressure measurement point setup which is $0.99\,kn$.

Fig. 4.2 shows how this combination of learning machines and input methods works. The forward / backward component of the flow velocity predicted by the combined system is shown on the ordinate and plotted against the actual flow velocity set in the simulations as for the original setup. This is done for all 2872 flow situations and the intensity of the diamonds indicates the number of results as before. Hence, a black diamond shows that there are a large number of results falling in this area.

As one can see quite a large number of results are very close to the perfect prediction (red line). However, there are some results that deviate significantly. So there are again problems with some unique flow situations that need to be addressed. It can also be observed that the linear fit of the data (green line) is very close to the perfect prediction as well which indicates that the overall performance is very good.

When compared to fig. 3.9 one may also observe that there are fewer outliers and the remaining

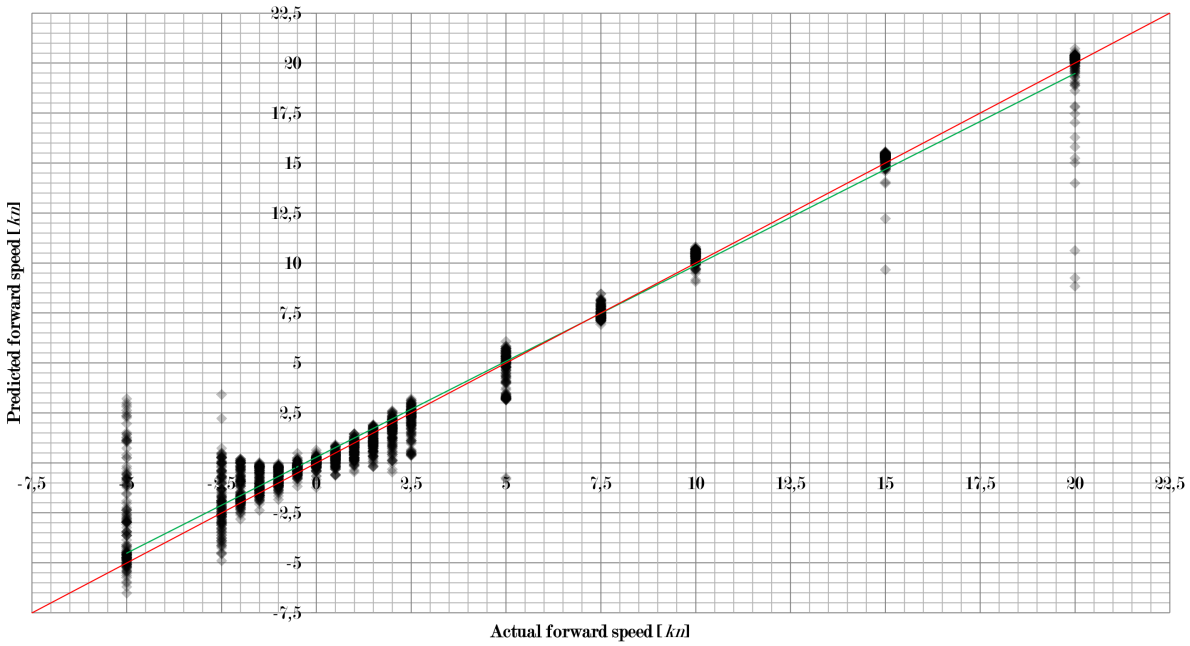


Figure 4.2: Predicted forward speed compared to actual forward speed (negative means backward).

points are closer to the ideal line. However, the remaining outliers are much further out than for the original setup. This can best be seen for the results at $5\,kn$, $15\,kn$, and $20\,kn$. There also seems to be more spread for high backward speed, especially at $-5\,kn$.

u $[kn]$	v $[kn]$	w $[kn]$	**Error in u**	
			Absolute $[kn]$	**Relative** $[\%]$
15.0	2.5	5.0	1.0	6.8
15.0	5.0	-5.0	5.3	35.6
15.0	5.0	2.5	2.8	18.6
20.0	-5.0	-5.0	10.8	53.8
20.0	-5.0	-2.5	1.0	5.2
20.0	-5.0	1.0	1.1	5.6
20.0	-5.0	1.5	2.5	12.7
20.0	-5.0	2.0	4.2	21.0
20.0	-5.0	2.5	6.0	30.0
20.0	-5.0	5.0	4.8	23.8
20.0	-2.5	-5.0	2.2	11.1
20.0	-2.5	5.0	9.4	46.9
20.0	1.5	5.0	1.4	6.9
20.0	2.0	5.0	3.0	14.9
20.0	2.5	-5.0	1.0	5.1
20.0	2.5	5.0	3.7	18.6
20.0	5.0	-5.0	5.0	24.9
20.0	5.0	-2.0	1.1	5.6
20.0	5.0	2.5	2.2	10.8
20.0	5.0	5.0	11.2	55.8

Table 4.2: Velocities which show the highest error for the forward component ($u \geq 7.5\,kn$).

As for the original setup it can be observed that all outliers produced in the range of $\pm 5\,kn$ for the forward velocity component are due to small sideward and upward / downward velocity

components of $\pm 1\,kn$ or less. So the change of the positions of the pressure measurement point has not removed this problem. However, the errors are smaller in this speed range with the exception of $-2.5\,kn$ and $-5\,kn$ where degradation in quality compared to the original setup can be seen.

For higher forward speeds of $7.5\,kn$ and above one can observe a different behaviour similar to the original setup. Here, the outliers occur for very specific combinations of velocity components as before. However, in contrast no strong outliers can be seen for $7.5\,kn$ and $10\,kn$. For these speeds the errors are reduced. Only $15\,kn$ and $20\,kn$ show a number of significant deviations. But also for these forward speeds there is a stronger concentration of points around the ideal line. Tab. 4.2 shows for which velocities the forward component is determined with the largest error. Again, no clear pattern can be observed with regard to the velocity components.

An improvement compared to the original setup is that now only two learning machines (support vector machines combined with raw and FFT input) are used. SVM with convoluted input is no longer required. Hence, less computational power is needed for the combined method.

4.2.2 Sideward Flow Velocity

Tab. 4.3 shows the root mean square errors (RMSE) in knots for all combinations of the 5 machine learning methods and the 5 different input data for the sideward velocity component.

RMSE [kn]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	2.26	-	0.37	-	0.84
FFT	2.26	-	0.35	-	0.84
Conv	2.21	-	0.89	-	1.09
MA	2.42	-	1.59	-	0.89
SVD	2.58	-	1.33	-	2.08

Table 4.3: Root means square errors for the sideward flow velocity.

As one can see no meaningful results can be obtained for k-nearest neighbour (KNN) and Bayesian networks (BN). For artificial neural networks (ANN) the RMSE is just below $2.5\,kn$ for all input methods except SVD data where the RMSE is little bit higher. For multiple linear regression (MLR) the root mean square error is between $0.8\,kn$ and $1.2\,kn$ except with singular value decomposition where it is just above $2\,kn$. Support vector machines (SVM) yield the lowest RSME when using raw data or Fourier transformed data (below $0.4\,kn$). With the other input methods the root mean square errors are between $0.8\,kn$ and $1.6\,kn$.

The overall behaviour of the machine learning algorithms deviates when compared to the forward / backward flow velocity calculations. The root mean square errors are lower and k-nearest neighbour and Bayesian networks do not work at all although BN yields good results for the forward / backward flow velocity. This could also be observed in the original setup.

When compared to the original results (tab. 3.3) the overall performance is again better. The reduction in the error is much more significant than for the forward / backward flow velocity and almost all input data / learning machine combinations show improvement. The exceptions are artificial neural networks with moving averages and singular value decomposition where there is a

slight increase in the RMSE. The other input methods do not yield much improvement for ANN and the overall results for this machine learning method are not as good as for the other ones.

Combined Method

In case of the sideward flow velocity all learning machine / input method combinations which yield meaningful results are of interest for the combined system. So ANN, SVM, and MLR with all input methods have to be taken into account to increase the accuracy and output stability of the whole system.

The best solution can be obtained using only support vector machines combined with raw and FFT input. The resulting root mean square error is $0.34\,kn$. This value is the lower than the lowest single method RMSE using SVM with FFT input. It is also 45 % lower than for the original pressure measurement point setup where the RMSE is $0.61\,kn$.

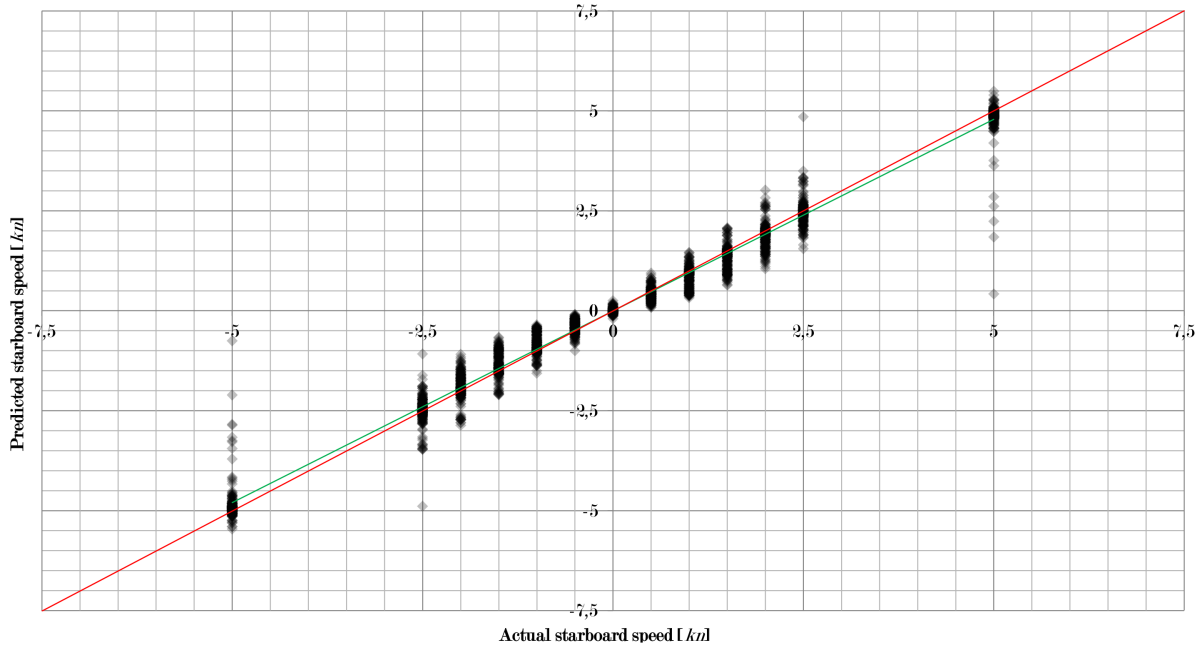


Figure 4.3: Predicted starboard speed compared to actual starboard speed (negative means port).

Fig. 4.3 shows how this combination of learning machines and input methods works. As before, the flow velocity predicted by the combined system is shown on the ordinate and plotted against the actual flow velocity set in the simulations.

As one can see quite a large number of results are very close to the ideal (red) line. Some deviations, but not many, can also be observed. So there are problems with some unique flow situations as before that need to be addressed. One may also observe that the linear fit of the data (green line) is closer to the perfect prediction than for the forward / backward case. So the overall performance is better for the sideward velocity component.

When compared to fig. 3.10 one can see that there are fewer outliers and the remaining points are much closer to the perfect prediction. The improvement is visualised quite well. Also the outliers are not as far out as before. In addition the linear fit of the data is much closer to the perfect fit compared to the original setup.

Having a closer look at all flow velocity components one may observe that in contrast to the original setup those points in 4.3 which deviate strongly from the ideal line no longer coincide with a large upward / downward component. Instead large errors in the determination of the sideward component occur for different sizes of the upward / downward. The same is true for the forward / backward component as before. Hence, the determination of the sideward component is now decoupled from the other flow velocity components.

Allowing a better distinction between the sideward and the upward / downward component of the flow velocity was one of the reasons to change the positions of the pressure measurement points. For the sideward component this has been achieved. In addition the relative error of the sideward component has been reduced by 17.7 % on average.

Another improvement compared to the original setup is that now only two learning machines (support vector machines combined with raw and FFT input) are used. Before, SVM with all input methods where required. This heavily reduced the computational power needed for the combined method.

4.2.3 Upward / Downward Flow Velocity

Tab. 4.3 shows the root mean square errors (RMSE) in knots for all combinations of the 5 machine learning methods and the 5 different input data for the upward / downward velocity component.

RMSE [kn]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	2.22	-	0.39	-	0.79
FFT	2.23	-	0.37	-	0.79
Conv	2.13	-	0.79	-	1.10
MA	2.26	-	1.43	-	0.85
SVD	2.55	-	1.26	-	1.94

Table 4.4: Root means square errors for the upward / downward flow velocity.

As one can see, again, no meaningful results can be obtained for k-nearest neighbour (KNN) and Bayesian networks (BN). For artificial neural networks (ANN) the RMSE is just below 2.5 kn for all input methods except for singular value decomposition (SVD) where the RMSE slightly above this value. For multiple linear regression (MLR) the root mean square error is around 1 kn except with singular value decomposition where it is just below 2 kn. Support vector machines (SVM) again yield the lowest RSME when using raw data or Fourier transformed data (below 0.4 kn). With the other input methods the root mean square error is between 0.7 kn and 1.5 kn.

The overall picture is very similar to sideward flow velocity situation. Again multiple linear regression gives good results and best results are obtained with raw data and FFT data. However, the performance gap between raw and FFT input compared to the other input methods is larger. While raw and FFT inputs yield even lower root mean square errors compared to the sideward flow velocity calculations, the performance of the other input methods is lower. This could also be observed in the original setup.

When compared to the original results (tab. 3.4) the overall performance is again better. The reduction in the error is not as significant as for the sideward flow velocity but all input data /

learning machine combinations show improvement. Not very much improvment can be observed for artificial neural networks with all input methods as well.

Combined Method

In case of the upward / downward flow velocity again all learning machine / input method combinations which yield meaningful results are of interest for the combined system as for the sideward flow velocity component. So ANN, SVM, and MLR with all input methods have to be taken into account.

The best solution can be obtained using only support vector machines combined with raw and FFT input. The resulting root mean square error is $0.37\,kn$. This value is the same as using SVM with FFT input. But it is lower than for the original pressure measurement point setup which is $0.40\,kn$.

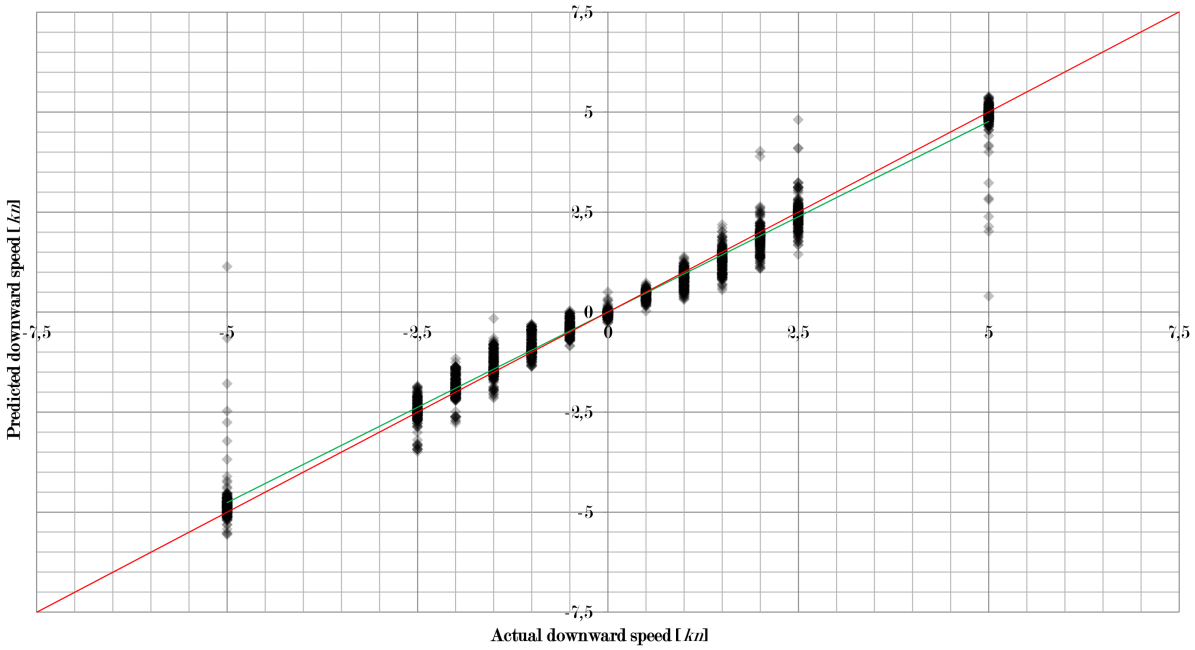


Figure 4.4: Predicted downward speed compared to actual downward speed (negative means upward).

Fig. 4.4 shows how the combination of learning machines and input methods works. Again, the flow velocity predicted is plotted against the actual flow velocity set in the simulations.

As one can see quite a large number of results are very close to the perfect prediction (red line). There are few results that deviate significantly. It can also be observed that the linear fit of the data (green line) is again very close to the perfect prediction.

When compared to fig. 3.11 one may observe that there are fewer outliers and the remaining points are much closer to the perfect prediction. It can also be seen that the outliers now tend to be "in to other direction", i.e. while in the original setup the velocity component is mostly underestimated it is more likely to be overestimated with the improved setup. The improvement is visualised quite well. Also the outliers are mostly not as far out as before with the exception of $-5\,kn$ where a single extreme outlier can be found. For flow velocity $\boldsymbol{u} = [-2.5\,kn, 0.0\,kn, -5.0\,kn]^T$ the upward / downward component is determined as $w_{estimate} = 1.14\,kn$ which gives a relative

error of 122.8 %.

Having a closer look at all flow velocity components it can be observed that in contrast to the original setup those points in fig. 4.4 which deviate strongly from the ideal line no longer coincide with a large sideward component. Instead large errors in the determination of the sideward component occur for different sizes of the sideward. The same is again true for the forward / backward component. Hence, the determination of the upward / downward component is now decoupled from the other flow velocity components as well.

Again only two learning machines (support vector machines combined with raw and FFT input) are used. In the original setup SVM with all input methods where required. So this heavily reduced the computational power which is needed for the combined method in this case as well.

5 Experimental Verification of Image Processing

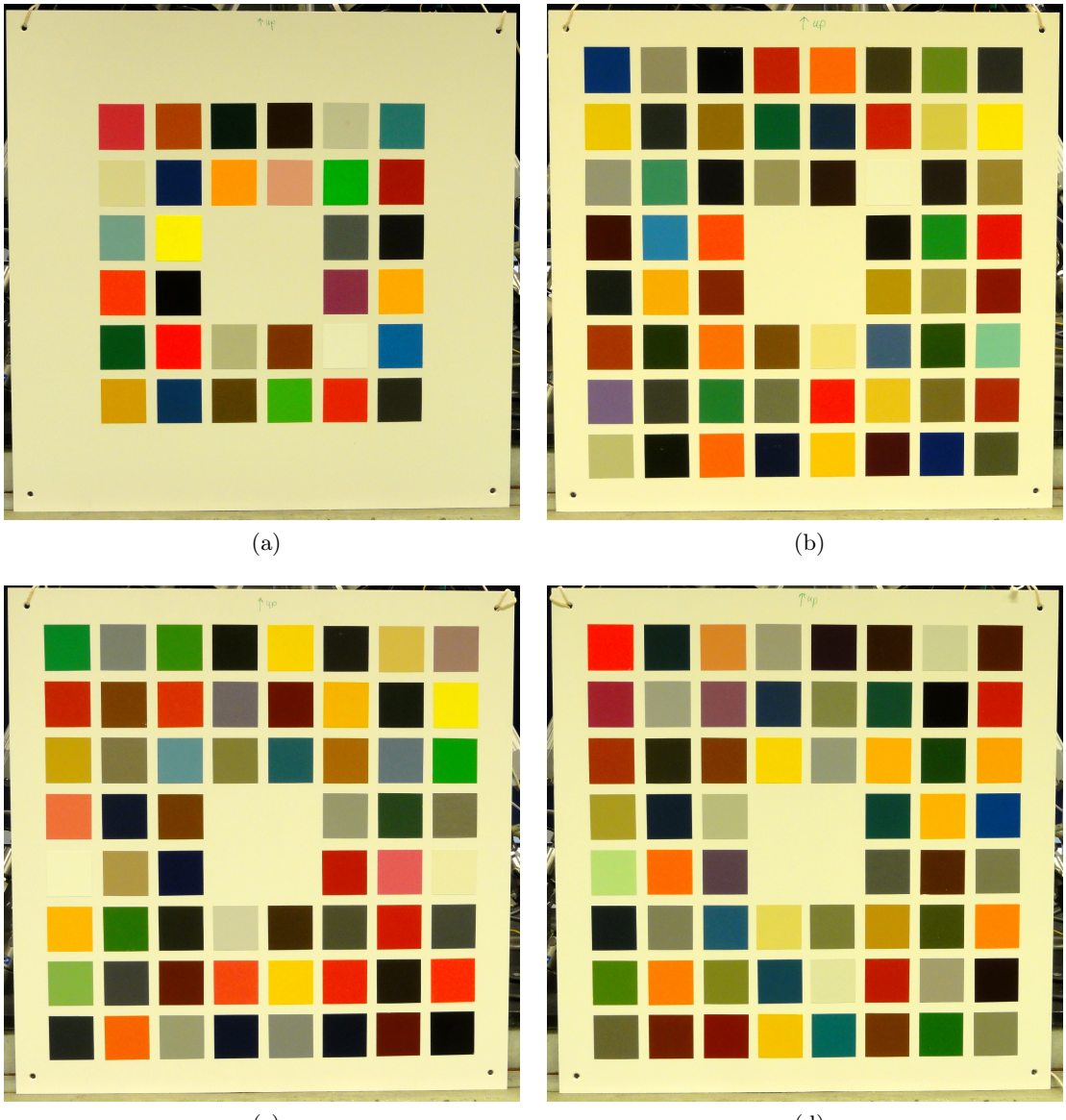
5.1 Experimental Setup and Procedure

5.1.1 Motivation

Up to now the colour transformation procedure was only evaluated in the controlled environment of a laboratory. It is of interest to test the applicability of learning algorithms for underwater image processing under more realistic conditions. This will allow a more appropriate evaluation of the method before it is implemented into a real system. Hence, an experimental verification in water is warranted.

5.1.2 Colour Patterns

In order to be able to evaluate the results as best as possible and to automate the colour matching for the learning algorithms the coloured objects from the laboratory test have to be replaced by



(a) (b)

(c) (d)

Figure 5.1: Colour patterns for the experimental verification.

colour patterns with known colours. For the colours the RAL colours are chosen which are 213 different colours covering a large range. RAL colour patches are widely available. Also the RGB values of the RAL colours are well-known such that the performance of the learning machines and the influence of the camera on the colours in the images can be evaluated.

As a base material for the colour patterns four white PVC boards are used. The boards are squared with a side length of 495 mm. On the boards a total of 212 squared colour patches (side length 40 mm) are placed. One of the 213 colours is left out such that the patches can be arranged in a regular grid for each pattern.

Fig. 5.1 shows the four colour patterns used for the experiment. The patterns are such that they can easily be processed with standard pattern recognition algorithms to extract the colours from the images at the appropriate positions. The first pattern (fig. 5.1(a)) consists of 32 carefully chosen colour patches. The selection is such that white, black, and several shades of grey are included as well as colours from the whole RGB range. The other three colour patters have 60 random patches. The first pattern is intended for training of the machine learning algorithms. The remaining patterns are for testing the resulting combined machine on various colours which are not part of the training process. In the following the patterns will be referred to as pattern 1 (top left), pattern 2 (top right), pattern 3 (bottom left), and pattern 4 (bottom right).

5.1.3 Procedure

The colour patterns are first photographed in the laboratory under white light at distances d_{cb} of 1 m, 5 m, and 8 m. This is done for various camera settings and different zooms to determine the best conditions. The camera used has an active pixel sensor (APS) with 10 Mpx resolution.

Figure 5.2: Diving basin at TauchRevierGasometer.

In the second step the boards are brought to Europe's largest indoor diving basin at TauchRevierGasometer in Duisburg. The diving basin is circular with a diameter of 45 m and a water depth

of 13 m. It provides conditions which are as close to open water as possible (see fig. 5.2) and it is therefore rated as open water for divers. Furthermore, the lighting conditions are such that greater water depths are simulated compared to the actual water depth achieved. Also the optical properties of the water are similar to coastal water (see fig. 1.7) and the water is very murky. So the penetration depth of light is expected to be very low in general with its maximum in the green colour region.

The boards are taken to depths d_w of 4 m, 8 m, and 12 m and again photographed at distances d_{cb} of 1 m, 5 m, and 8 m under different lighting condition using the same camera that is used in the laboratory. As under white light this is done for various camera settings and different zooms. Fig. 5.3 shows a sketch of the experiment.

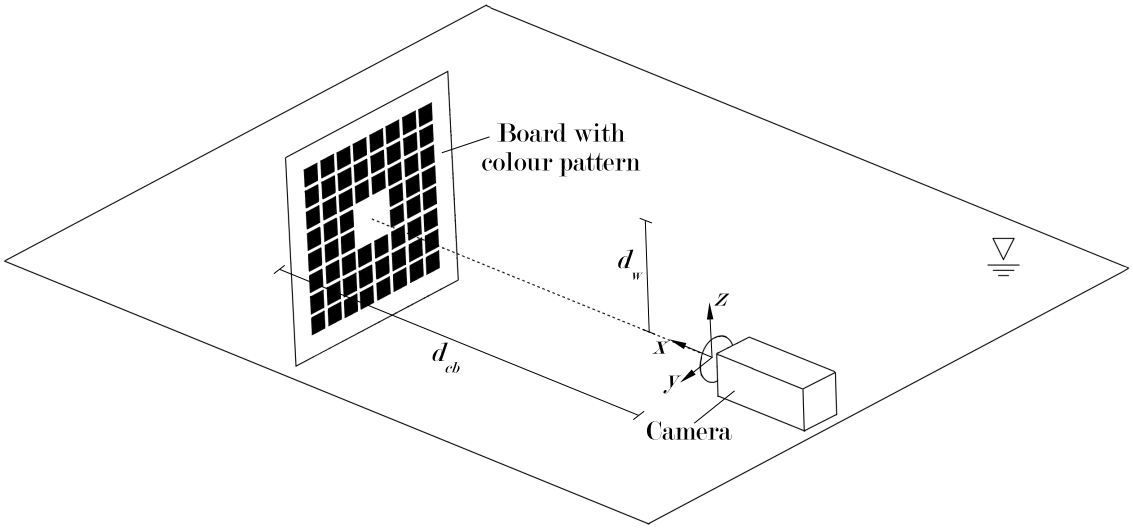


Figure 5.3: Experimental setup.

The images taken are then processed as follows. First pattern recognition algorithms are used to find the positions of the colour patches in the images. The patches from every underwater image are matched with those from a corresponding image taken in the laboratory under white light at the same distance and with the same camera settings.

The colours are then extracted from the pixels in the patches and the data is again stored in an $n \times 6$ matrix containing the colour channels (RGB) for two matching pixels in each row. Finally, the order of the rows is again randomised. The resulting data matrix is used as input for the machine learning software EIDOminer.

5.2 Results and Discussion

5.2.1 Some Remarks on Performing the Underwater Tests

Due to the conditions on site it was not possible to obtain images of every colour pattern for every water depth and camera distance. In addition not all combinations of distance and depth gave feasible results. Tab. 5.1 shows an overview of the performance.

No feasible images could be obtained at a water depth of 12 m and a camera distance of 5 m. Fig. 5.4 shows an image taken of pattern 2 under these conditions. As one can see only the measurement cord is visible in the left part of the image. The light cannot penetrate the water

		Camera distance [m]		
		1	5	8
Water depth [m]	4	all patterns	all patterns	pattern 1
	8	patterns 2, 3, and 4	patterns 2, 3, and 4	no images
	12	patterns 1 and 2	no feasible images	no images

Table 5.1: Performance of the underwater tests.

to the colour pattern itself. The same is true for a camera distance of $8\,m$ and water depths of $8\,m$ and $12\,m$. Hence, no images were obtained for these conditions.

Figure 5.4: Pattern 2 at a water depth of $12\,m$ and a camera distance of $5\,m$.

Tab. 5.1 also shows that pattern 1 is missing for a water depth of $8\,m$ and camera distances $1\,m$ and $5\,m$. As pattern 1 is intended as the calibration pattern other colours from the remaining patterns need to be chosen for calibration. In order to obtain comparable results colours which are as close as possible to the colours in pattern 1 are selected. Tab. 5.2 gives the replacement colours arranged with respect to colour pattern 1. For each entry the top number indicates the colour pattern and the two numbers below give the row and column of the selected colour. Further processing of the data is then done as for the other conditions but with the replacement colours instead of pattern 1 for training of the machine learning algorithms.

$\frac{3}{5,7}$	$\frac{4}{3,1}$	$\frac{2}{6,7}$	$\frac{4}{1,6}$	$\frac{4}{1,7}$	$\frac{4}{8,5}$
$\frac{2}{6,5}$	$\frac{2}{8,7}$	$\frac{4}{6,8}$	$\frac{3}{5,7}$	$\frac{3}{3,8}$	$\frac{3}{5,6}$
$\frac{3}{7,1}$	$\frac{3}{2,8}$			$\frac{3}{6,8}$	$\frac{2}{2,2}$
$\frac{3}{7,6}$	$\frac{4}{2,7}$			$\frac{4}{2,1}$	$\frac{3}{7,5}$
$\frac{2}{6,7}$	$\frac{3}{7,8}$	$\frac{3}{6,4}$	$\frac{4}{3,3}$	$\frac{2}{3,6}$	$\frac{2}{1,1}$
$\frac{4}{3,4}$	$\frac{2}{8,7}$	$\frac{2}{6,4}$	$\frac{3}{1,3}$	$\frac{2}{7,5}$	$\frac{3}{6,6}$

Table 5.2: Replacement colours for missing pattern 1.

5.2.2 Red Channel

Tab.'s 5.3 to 5.8 show the root mean square errors (RMSE) for all combinations of the 5 machine learning methods and the 5 different input data for the red channel of 24 bit RGB images, i.e. every channel has 256 possible pixel values. As mentioned before the actual deviation of a pixel

colour from the expected value can deviate significantly.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	75.172	15.69	19.82	62.31	45.07
FFT	74.61	16.31	19.66	61.49	45.07
Conv	74.42	16.27	19.36	62.16	45.08
MA	74.35	73.38	41.69	62.11	45.07
SVD	74.60	15.69	19.82	62.31	45.07

Table 5.3: RMSEs for the red channel at a water depth of $4\,m$ and a camera distance of $1\,m$.

Tab. 5.3 gives the results for a water depth of $4\,m$ and a camera distance of $1\,m$. As one can see artificial neural networks (ANN), Bayesian networks (BN), and multiple linear regression (MLR) show large RMSEs independent of the input data. In fact the RMSE varies less than $1\,px$ for the different input methods. In contrast, for k-nearest neighbour (KNN) and support vector machines (SVM) the performance depends on the input data. But the performance is much better than for the other three learning methods with root means square errors between $15\,px$ and $20\,px$ compared to more than $45\,px$ for MLR, $62\,px$ for BN, and $75\,px$ for ANN. The exception is moving averages (MA) where the RMSE is above $42\,px$ for SVM and more than $73\,px$ for KNN.

The performance pattern is very similar to the laboratory results with a low performance of Bayesian networks, multiple linear regression, and artificial neural networks. For the other methods we have very small RMSE in contrast to the original results. This is to be expected as the camera distance is only $1\,m$ and the red colour is not as strongly filtered in coastal water as compared to ocean water. This can be seen in fig. 5.5. The colours of the patches are distinctively different although there is a greenish tint.

Figure 5.5: Pattern 1 at a water depth of $4\,m$ and a camera distance of $1\,m$.

An interesting observation can again be made about smoothing the input data with moving averages. In the original setup this reduces the performance of the learning machines especially for KNN and SVM. Here the effect is exactly the same for SVM and KNN. However, for the other learning methods the drop cannot be observed.

Tab. 5.4 shows the results for a water depth of $4\,m$ and a camera distance of $5\,m$. The results are very similar to $1\,m$ camera distance. For most machine learning / input method combinations a

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	76.48	15.47	19.94	62.32	45.07
FFT	77.01	16.24	19.75	62.35	45.07
Conv	76.82	16.24	20.15	61.53	45.08
MA	77.49	72.33	63.94	62.02	45.07
SVD	76.87	15.47	19.94	62.32	45.07

Table 5.4: RMSEs for the red channel at a water depth of $4\,m$ and a camera distance of $5\,m$.

very slight increase of the root mean square error can be observed. The most notable exception is k-nearest neighbour where the performance increases a little bit independent of the input method. But the overall pattern with a low performance of ANN, BN, and MLR compared to good results for KNN and SVM (excluding MA) is the same. It may therefore be concluded that the additional $4\,m$ light has to travel from the colour patterns to the camera do not results in much stronger filtering due to water and submerged particles. Also the water depth is not very large such that the ambient lighting is quite good.

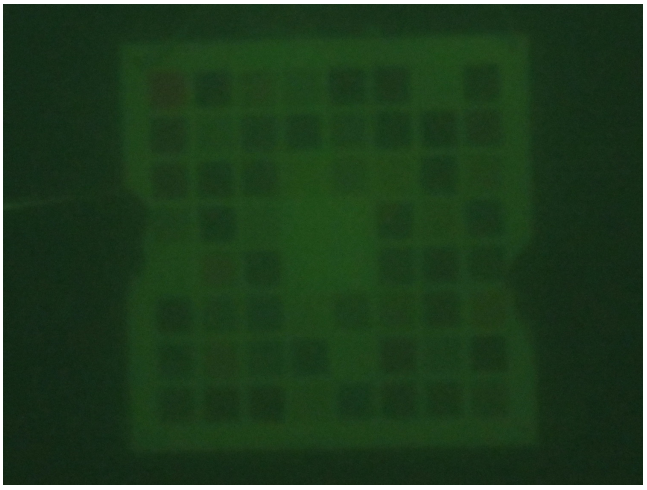


Figure 5.6: Pattern 4 at a water depth of $4\,m$ and a camera distance of $5\,m$.

Fig. 5.6 shows pattern 4 under these conditions. As before one can distinguish the colours. However, there is a greenish tint that is much stronger than for a camera distance of $1\,m$. In addition the image is quite hazy.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	76.32	-	62.02	67.67	58.55
FFT	76.32	-	59.38	67.11	58.55
Conv	76.31	-	60.06	66.81	58.57
MA	75.28	-	63.20	65.62	58.57
SVD	76.35	-	62.02	67.67	58.55

Table 5.5: RMSEs for the red channel at a water depth of $4\,m$ and a camera distance of $8\,m$.

Tab. 5.5 shows the results for a water depth of $4\,m$ and a camera distance of $8\,m$. Significant changes in the performances can be observed compared to the results for $1\,m$ and $4\,m$ camera distance. While the already poor quality of BN is only slightly reduced further and ANN show almost no change, a very strong drop in performance can be seen for SVM and MLR. The RMSE remains mainly independent of the input method for MLR but is now increased to around $59\,px$

compared to just above 45 px at 4 m camera distance. For KNN no meaningful results could be obtained and for SVM the RMSE is now between 59 px and 62 px such that SVM produces about the same quality as MLR. Again SVM with MA sticks out with significantly higher RMSEs. The results show that it is much more difficult to accurately reconstruct the colour of the patches from the underwater images then for the previous two conditions.

Figure 5.7: Pattern 1 at a water depth of 4 m and a camera distance of 8 m.

This can also be seen in fig. 5.7 where pattern 1 is shown at 4 m depth and 8 m camera distance. The image is very dark and it is very difficult to separate the colours.

RMSE [px]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	73.92	14.63	21.86	66.37	52.06
FFT	73.82	15.07	18.52	66.42	52.06
Conv	74.23	14.71	20.04	66.41	52.07
MA	77.36	67.11	63.78	61.15	52.14
SVD	73.86	14.63	21.86	66.37	52.06

Table 5.6: RMSEs for the red channel at a water depth of 8 m and a camera distance of 1 m.

Tab. 5.6 shows the results for a water depth of 8 m and a camera distance of 1 m. The performance of the learning machines can be directly compared to the results for 4 m depth and 1 m camera distance. For Bayesian networks (BN) and multiple linear regression (MLR) the performance is slightly worse than for 4 m depth. However, for artificial neural networks (ANN), k-nearest neighbour (KNN), and support vector machines (SVM) the results are - surprisingly - slightly better in most cases. The RSMEs are lower by up to 2 px.

The results can be explained as follows. The amount of ambient light has only little influence on the performance in this situation as the lighting is already very low as shown in fig. 5.2. The main factor of influence is the distance the light has to travel from the colour patterns to the board which is 1 m in both cases. Hence, the results for 4 m and 8 m are very similar. The slightly better performance at 8 m for some of the algorithms may be due to marginally better conditions in the surroundings when the image was taken. Other explanations are possible as well. There is no clear single reason that can be given.

Tab. 5.7 shows the results for a water depth of 8 m and a camera distance of 5 m. In contrast to the previous situation a direct comparison to the same camera distance at 4 m water depth

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	90.72	-	53.95	72.29	65.49
FFT	90.71	-	57.73	72.29	65.49
Conv	90.54	-	54.76	72.28	65.49
MA	86.13	-	80.50	73.41	65.50
SVD	90.85	-	53.95	72.29	65.49

Table 5.7: RMSEs for the red channel at a water depth of $8\,m$ and a camera distance of $5\,m$.

cannot be done. In fact, the results are more similar to $8\,m$ camera distance at $4\,m$ with slightly better performance of SVM. So the combination of camera distance and reduction of ambient lighting seems to result in a significant degradation which was not apparent at $4\,m$ water depth. It may also be observed that KNN does not produce meaningful results for this condition.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	81.93	-	48.06	63.59	51.25
FFT	81.92	-	48.74	63.07	51.25
Conv	81.42	-	49.56	62.60	51.26
MA	80.50	-	72.10	62.94	51.29
SVD	81.46	-	53.37	63.59	51.25

Table 5.8: RMSEs for the red channel at a water depth of $12\,m$ and a camera distance of $1\,m$.

The results for $12\,m$ water depth and $1\,m$ (tab. 5.8) confirms this. Although the camera distance is small the performance of the learning algorithms is reduced compared to $4\,m$ and $8\,m$ water depth. So the amount of ambient light also has an influence. However the distance of the object from the camera shows more significance toward the root means square errors. KNN also produces no meaningful results at this depth.

Combined Method

Again, as for the analysis of the laboratory results it is not feasible to test all possible learning machine and input method combinations for the increase of accuracy and output stability. Tab. 5.3 indicates that KNN and SVM with all input methods except moving averages are of interest for further study of the red channel at $4\,m$ depth and $1\,m$ camera distance.

The best solution can be obtained using only k-nearest neighbour combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is $14.39\,px$. This improves the best result obtained with a single machine (KNN with Raw or SVD) by more than $1\,px$.

Fig. 5.8 shows how this combination of learning machines and input methods works. The red value of a pixel predicted by the combined system is shown on the ordinate and plotted against the actual red value of the pixel obtained from an image under white light. This is done for a large sample of pixels (about 12800) taken from different test images. As before, the intensity of the diamonds indicates the number of results. Hence, a black diamond shows that there are a large number of results falling in this area.

As one can see quite a large number of results are very close to the perfect prediction (red line). There are some results that deviate. But these are only a small number and a clear trend and clustering around the red line can be seen. It can also be observed that the linear fit of the

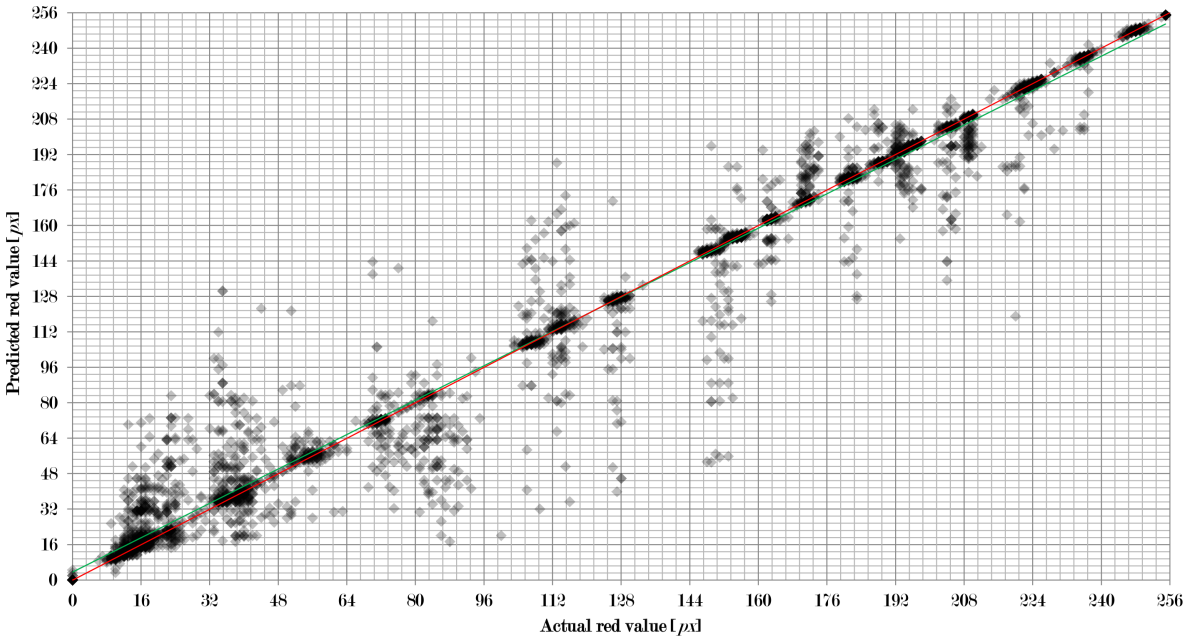


Figure 5.8: Predicted red value compared to actual red value at a water depth of $4\,m$ and a camera distance of $1\,m$.

data (green line) is almost on top of the line of perfect prediction indicating that the overall performance is excellent.

Fig. 5.8 also shows that in contrast to the laboratory results there is no increase of the error with increasing red value. This is not the case as the colours on the board are distinguishable under these conditions. Hence, the reconstruction of the colour channel is much easier than in the laboratory (see fig. 5.6). However, the RSME is much higher than for the laboratory which is $5.27\,px$. This is due to the very few but quite significant outliers as shown in fig. 5.8. In the lab there are more pixels deviating from the ideal line but the errors of each are smaller (see fig. 3.12).

In case of $5\,m$ camera distance at $4\,m$ water depth again KNN and SVM with all input methods except moving averages are of interest for further study of the red channel. As for $1\,m$ camera distance the best solution can be obtained using only k-nearest neighbour combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is $14.47\,px$. This improves the best result obtained with a single machine by about $1\,px$. The improvement is slightly smaller than for $1\,m$ camera distance.

Fig. 5.9 shows how this combination of learning machines and input methods works. The results look very similar to those for $1\,m$ camera distance. Even the outliers are situated in the same regions in the diagram. The results of the single learning machine / input method combinations (tab. 5.4) were already very similar to the $1\,m$ camera distance results. Hence, it can be expected that the combined methods are also closely related.

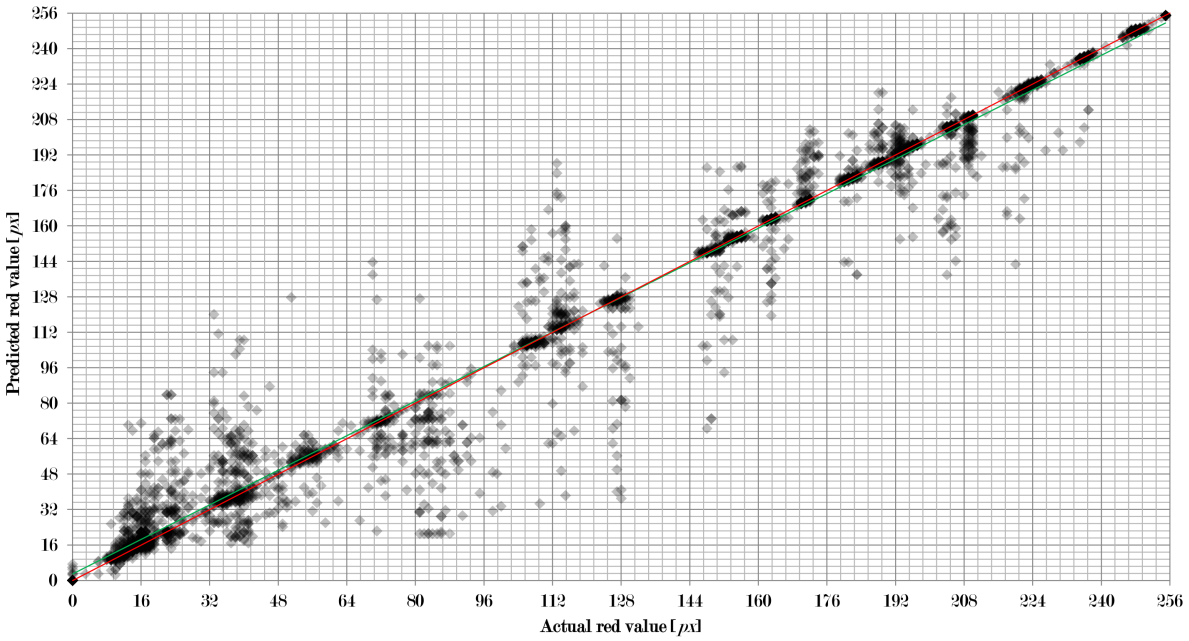


Figure 5.9: Predicted red value compared to actual red value at a water depth of $4\,m$ and a camera distance of $5\,m$.

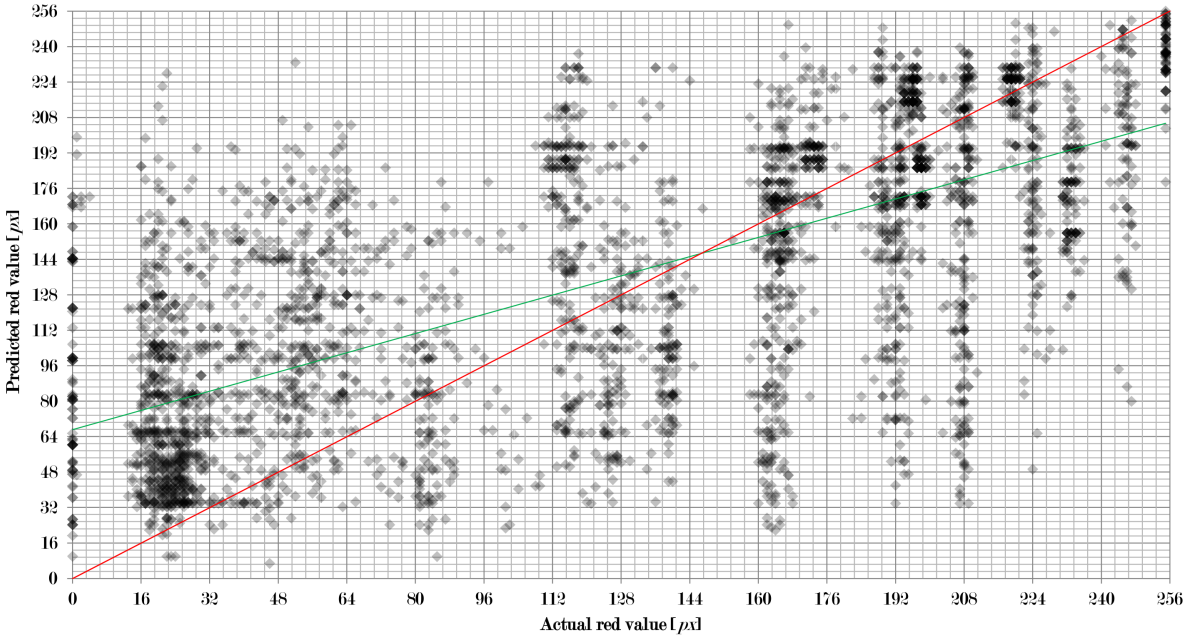


Figure 5.10: Predicted red value compared to actual red value at a water depth of $4\,m$ and a camera distance of $8\,m$.

In case of $8\,m$ camera distance and $4\,m$ water depth SVM with all input methods except moving averages are of interest for further study of the red channel as before. In addition multiple linear regression also shows promise in this case as the RMSE is not very high for all input methods. In contrast to the previous two conditions k-nearest neighbour cannot be used.

The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. Multiple linear regression is not considered. The resulting root mean square error is $58.55\,px$. This is slightly better than MLR which gives the best single method errors.

Fig. 5.10 shows how this combination of learning machines and input methods works. As for the single methods significant changes in the performances can be observed compared to the results for 1 m and 4 m camera distance. The points are widely spread out. However, a closer observation reveals that many points are clustered near the ideal line. It can also be seen that for smaller pixel values the red value is more likely to be overestimated and for higher pixel values to be underestimated. This is sensible as the range of pixel values in the underwater images is smaller than the complete range available under white light.

Tab. 5.6 indicates that KNN and SVM with all input methods except moving averages are of interest for further study of the red channel at 8 m depth and 1 m camera distance. The best solution can be obtained using only k-nearest neighbour combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is 13.01 px. This improves the best result obtained with a single machine (KNN with Raw or SVD) by more than 1 px.

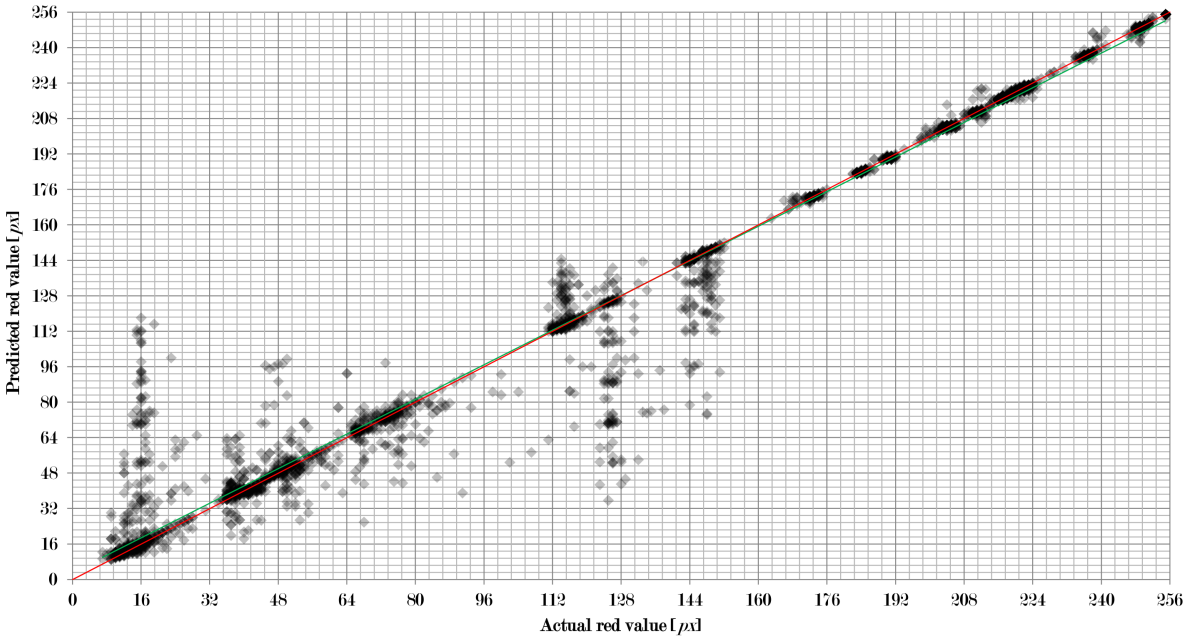


Figure 5.11: Predicted red value compared to actual red value at a water depth of 8 m and a camera distance of 1 m.

Fig. 5.11 shows how this combination of learning machines and input methods works. The observeration that the results for 8 m are better than for 4 m depth which was made for the single methods is also apparent with the combined method when comparing fig. 5.8 and fig. 5.11. There is almost no error for pixel values above 160 px and for lower values the outliers are few.

In case of 5 m camera distance and 8 m water depth SVM with all input methods except moving averages are of interest for further study of the red channel as before. In addition multiple linear regression also shows promise in this case as the RMSE is not very high for all input methods. k-nearest neighbour cannot be used as no meaningful results are available.

The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. Multiple linear regression is not considered. The resulting root mean square error is 52.91 px. This is more than 1 px lower than the RSME of SVM with Raw or SVD input which give the lowest single method errors.

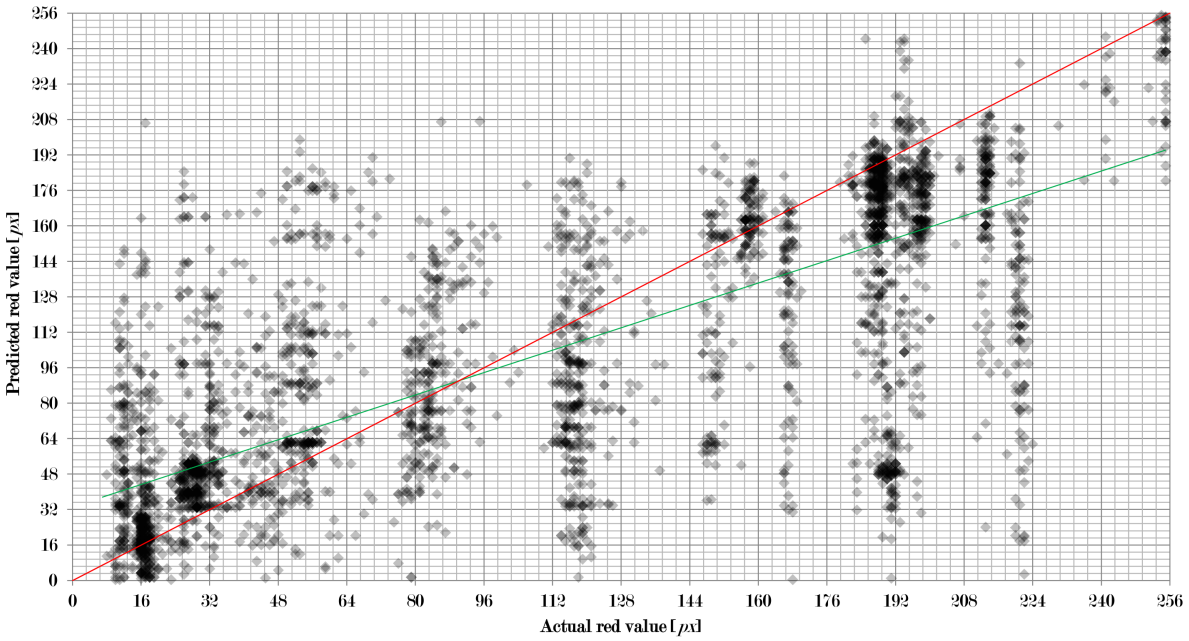


Figure 5.12: Predicted red value compared to actual red value at a water depth of 8 m and a camera distance of 5 m.

Fig. 5.12 shows how this combination of learning machines and input methods works. The points are more spread out than for 1 m camera distance. A qualitative similarity to the results at 4 m water depth and 8 m camera distance can be seen. However, the results shown in fig. 5.12 are much better. One can also see that the points are well-clustered near the red line. As for 4 m water depth and 8 m camera distance the red value is more likely to be overestimated for smaller pixel values and underestimated for higher pixel values.

In case of 1 m camera distance and 12 m water depth SVM with all input methods except moving averages are of interest for further study of the red channel as before. In addition multiple linear regression also shows promise in this case as the RMSE is not very high for all input methods. Again, k-nearest neighbour cannot be used.

The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. Multiple linear regression is not considered. The resulting root mean square error is 47.37 px. This is almost 1 px lower than the RSME of SVM with Raw input which give the lowest single method errors.

Fig. 5.13 shows how this combination of learning machines and input methods works. Again, the points are spread out but even less than for 5 m camera distance and 8 m water depth. Also the overestimation of small and the underestimation of high red values is not as apparent as for 8 m camera distance at 4 m water depth and 5 m camera distance at 8 m water depth.

5.2.3 Green Channel

Tab.'s 5.9 to 5.14 show the root mean square errors (RMSE) for all combinations of the 5 machine learning methods and the 5 different input data for the green channel of 24 bit RGB images.

Tab. 5.9 gives the results for a water depth of 4 m and a camera distance of 1 m. As one can see artificial neural networks (ANN) show a large RMSE independent of the input data just

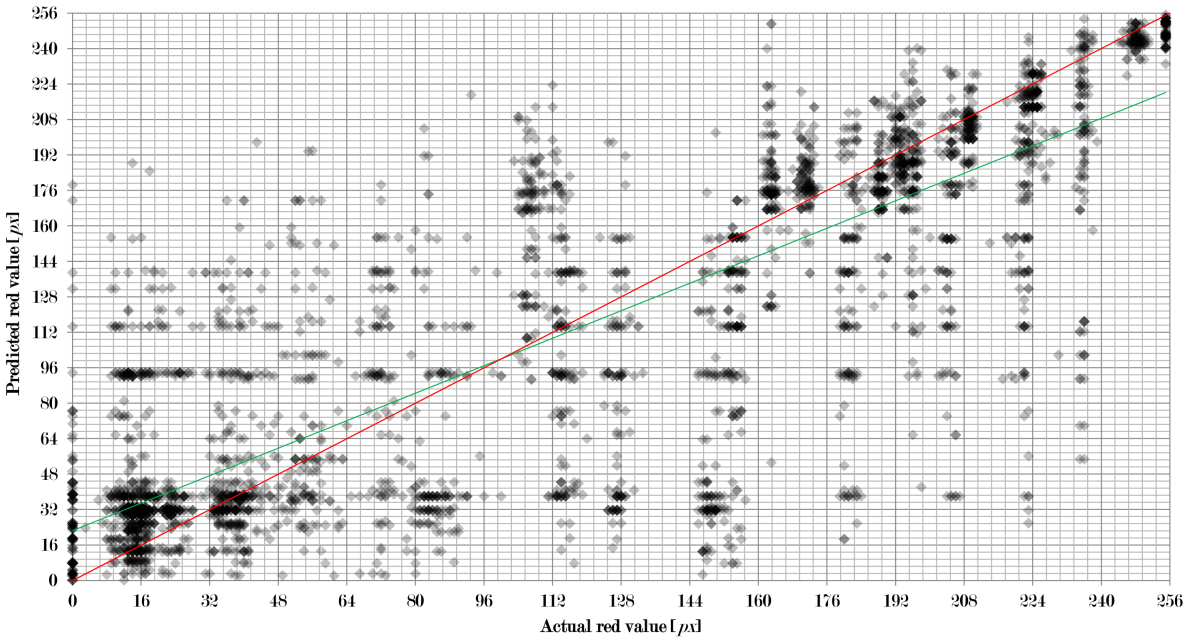


Figure 5.13: Predicted red value compared to actual red value at a water depth of 12 m and a camera distance of 1 m.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	62.54	12.97	12.52	34.13	21.86
FFT	62.60	13.45	12.21	33.86	21.86
Conv	62.52	13.66	12.93	33.64	21.87
MA	62.19	67.74	51.61	33.99	21.88
SVD	62.61	12.97	12.52	34.13	21.86

Table 5.9: RMSEs for the green channel at a water depth of 4 m and a camera distance of 1 m.

above 62 px. The other machine learning algorithms show a much smaller RMSEs except for support vector machines (SVM) and k-nearest neighbour (KNN) combined with moving averages (MA) where the errors are in a similar range. For multiple linear regression (MLR) the RSME is below 22 px and also varies very little with respect to the input data. The same is true for Bayesian networks (BN) where the RSMEs are around 34 px. In contrast, for KNN and SVM the performance depends on the input data. These methods show root mean square errors between 12 px and 14 px. Best results are obtained with SVM and FFT with a root mean square error of just above 12 px.

As for the red channel the performance pattern is very similar to the laboratory results with a low performance of Bayesian networks, multiple linear regression, and artificial neural networks. For the other methods we have similar RMSEs. Also the performance is much better than for the red channel under this condition. This is to be expected as green light has the highest penetration depth in costal water as shown in fig. 1.7.

Tab. 5.10 shows the results for a water depth of 4 m and a camera distance of 5 m. As for the red channel, the results are very similar to 1 m camera distance. With BN and MLR the root mean square errors show almost no change. A slight increase in performance can be observed for ANN and for KNN and SVM there are very slight fluctuation in performance of less than 1 px. From these results the same conclusions as for the red channel can be drawn: the additional 4 m light

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	61.69	12.79	12.28	34.13	21.86
FFT	61.61	13.40	12.93	33.79	21.86
Conv	61.63	13.28	12.19	33.71	21.87
MA	62.72	66.64	45.61	33.21	21.87
SVD	61.45	12.79	12.28	34.13	21.86

Table 5.10: RMSEs for the green channel at a water depth of $4\,m$ and a camera distance of $5\,m$.

has to travel from the colour patterns to the camera do not results in much stronger filtering due to water and submerged particles.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	64.17	-	40.31	49.41	40.71
FFT	64.20	-	40.45	47.31	40.71
Conv	64.32	-	40.70	46.24	40.83
MA	59.72	-	40.72	47.63	40.72
SVD	64.31	-	40.31	49.41	40.71

Table 5.11: RMSEs for the green channel at a water depth of $4\,m$ and a camera distance of $8\,m$.

Tab. 5.11 shows the results for a water depth of $4\,m$ and a camera distance of $8\,m$. As for the red channel significant changes in the performances can be observed compared to the results for $1\,m$ and $5\,m$ camera distance except for ANN where there are very little changes of the RMSEs. For MLR the drop in performance is around $19\,px$ independent of the input method and for BN the increase of the RMSE is up to $16\,px$. The strongest drop in performance can be seen for KNN and SVM. For KNN no meaningful results are obtained and for SVM the RMSE is now between $40\,px$ and $41\,px$ such that SVM produce about the same quality as MLR. So also for the green channel it is much more difficult to reconstruct the colour of the patches from the underwater images accurately than for the previous two conditions.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	62.66	22.24	22.46	41.55	38.57
FFT	62.34	22.84	21.30	41.32	38.57
Conv	62.56	22.71	21.33	41.34	38.58
MA	64.41	68.36	47.12	46.01	38.58
SVD	62.58	22.24	22.46	41.55	38.57

Table 5.12: RMSEs for the green channel at a water depth of $8\,m$ and a camera distance of $1\,m$.

Tab. 5.12 shows the results for a water depth of $8\,m$ and a camera distance of $1\,m$. In contrast to the results for the red channel the performance of the learning machines cannot be directly compared to the results for $4\,m$ depth and $1\,m$ camera distance. With the exception of ANN the results are significantly worse than for $4\,m$ water depth. This is especially notable for KNN, SVM, and MLR. The RMSE's for MLR are now above $38\,px$. For KNN and SVM the RMSEs vary between $21\,px$ and $23\,px$. In fact for KNN these are higher values than for the red channel under the same conditions. For SVM the RMSE's for red and green channel are about the same under these conditions.

As one can see the explanation for the behaviour of the performance of the learning algorithms at

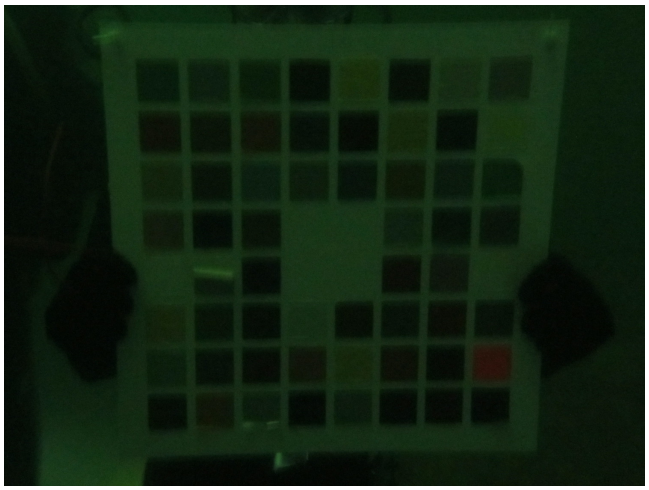


Figure 5.14: Pattern 3 at a water depth of $8\,m$ and a camera distance of $1\,m$.

$8\,m$ water depth and $1\,m$ camera distance given for the red channel is not sufficient when looking at the results for the green channel. The main factor of influence is still the distance the light has to travel from the colour patterns to the board. However, the optical properties of the suspended particles in the water also have to be taken into account. As the water is quite murky these have a significant influence on the results. It can be expected that the water in the diving basin has a high content of rust particle due to the steel encasing. Rust appears mainly red. Hence, all colours except red are absorbed stronger. This would lead to more absorption in the green channel (see fig. 5.14) and hence the results observed.

RMSE [px]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	61.66	39.65	32.59	55.18	51.84
FFT	61.69	41.16	31.58	55.18	51.84
Conv	61.44	42.43	32.04	55.18	51.84
MA	67.22	79.40	58.59	54.11	51.86
SVD	62.41	39.65	32.59	55.18	51.84

Table 5.13: RMSEs for the green channel at a water depth of $8\,m$ and a camera distance of $5\,m$.

Tab. 5.13 shows the results for a water depth of $8\,m$ and a camera distance of $5\,m$. As for the red channel the results are similar to $8\,m$ camera distance at $4\,m$. But compared to the results for the red channel under the same conditions the performance is much better. One now has RMSEs around $62\,px$ for ANN, around $55\,px$ for BN, and just below $52\,px$ for MLR. The root mean square errors for KNN and SVM are again lower compared to the other three methods and have more variation with respect to the input method between $31\,px$ and $41\,px$. So the combination of camera distance and reduction of ambient lighting results in a significant degradation in performance.

In fig. 5.15 one can clearly see the strong reduction of light. In addition it can be observed that the image is quite hazy and has a strong greenish connotation.

Finally, tab. 5.14 shows the results for a water depth of $12\,m$ and a camera distance of $1\,m$. The performance of the learning machines can be directly compared to the results for $8\,m$ depth and $1\,m$ camera distance. The results are about the same for ANN and KNN. But the performance of SVM, BN, and MLR is increased notably compared to $8\,m$ water depth.

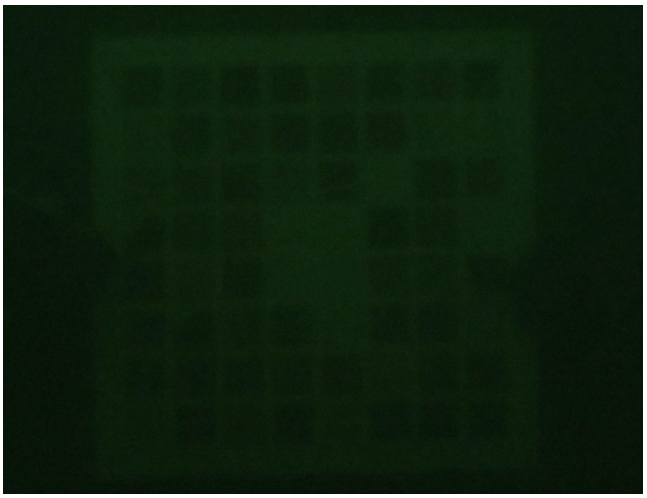


Figure 5.15: Pattern 2 at a water depth of 8 m and a camera distance of 5 m.

RMSE [px]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	63.23	22.89	20.60	38.63	24.99
FFT	63.32	24.32	21.01	36.97	24.99
Conv	62.90	25.23	20.89	36.67	25.00
MA	66.33	70.41	36.64	34.40	25.02
SVD	63.04	22.89	20.60	38.63	24.99

Table 5.14: RMSEs for the green channel at a water depth of 12 m and a camera distance of 1 m.

The results confirm the assumption that the camera distance has a greater influence on the performance than water depth as the performance is quite good. However, when compared to the other results at 1 m camera distance the increase of RMSE with depth does not hold here. One would expect much greater errors. But when looking at one of the images taken under these conditions (fig. 5.16) one may observe that the colour pattern is very clear and sharp. This indicates that there are fewer particles in the water and therefore less absorption of light between pattern and camera.

Combined Method

As before, it is not feasible to test all possible learning machine and input method combinations for the increase of accuracy and output stability. Tab. 5.9 indicates that KNN and SVM with all input methods except moving averages are of interest for further study of the green channel at 4 m depth and 1 m camera distance.

The best solution can be obtained using k-nearest neighbour and support vector machines combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is 9.64 px. This improves the best result obtained with a single machine (SVM with FFT) by almost 3 px.

Fig. 5.17 shows how this combination of learning machines and input methods works. The green value of a pixel predicted by the combined system is shown on the ordinate and plotted against the actual green value of the pixel obtained from an image under white light. This is done for a large sample of pixels (about 12800) taken from different test images. As before, the intensity of the diamonds indicates the number of results.

As one can see quite a large number of results are very close to the perfect prediction (red line).

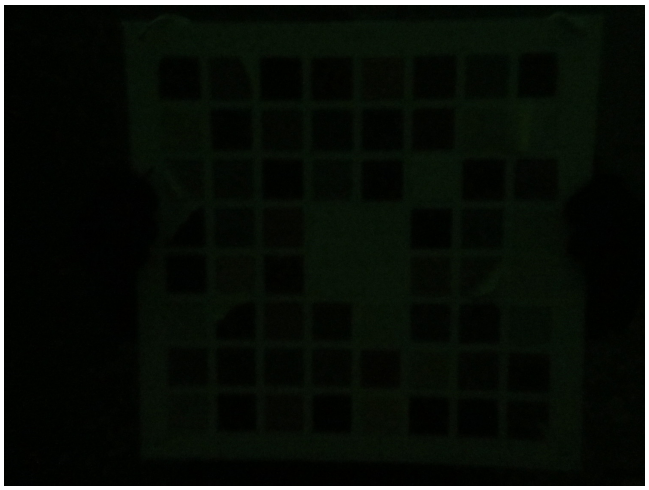


Figure 5.16: Pattern 2 at a water depth of 12 m and a camera distance of 1 m.

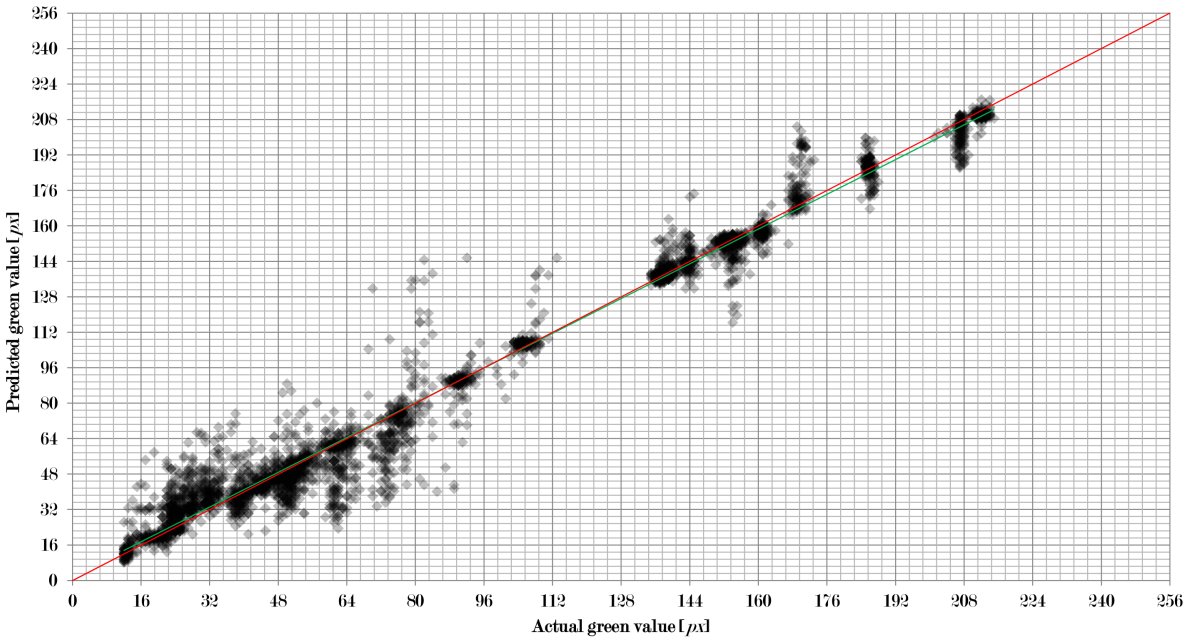


Figure 5.17: Predicted green value compared to actual green value at a water depth of 4 m and a camera distance of 1 m.

There are some results that deviate. But these are only a small number and a clear trend and clustering around the red line can be seen.

A comparison to the results for the red channel (fig. 5.8) shows that there are more points which do not lie on the ideal line for the green channel. However, there are significantly fewer outliers. Furthermore, the number of outliers decreases with increasing pixel value.

Compared to the results of the laboratory experiment the resulting root mean square error is higher than for the laboratory as for the red channel. But the overall structure is very similar for the green channel.

In case of 5 m camera distance at 4 m water depth again KNN and SVM with all input methods except moving averages are of interest for further study of the green channel. The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is 11.91 px. This improves the best result

obtained with a single machine by less than 1 px. The improvement is not as significant as for 1 m camera distance.

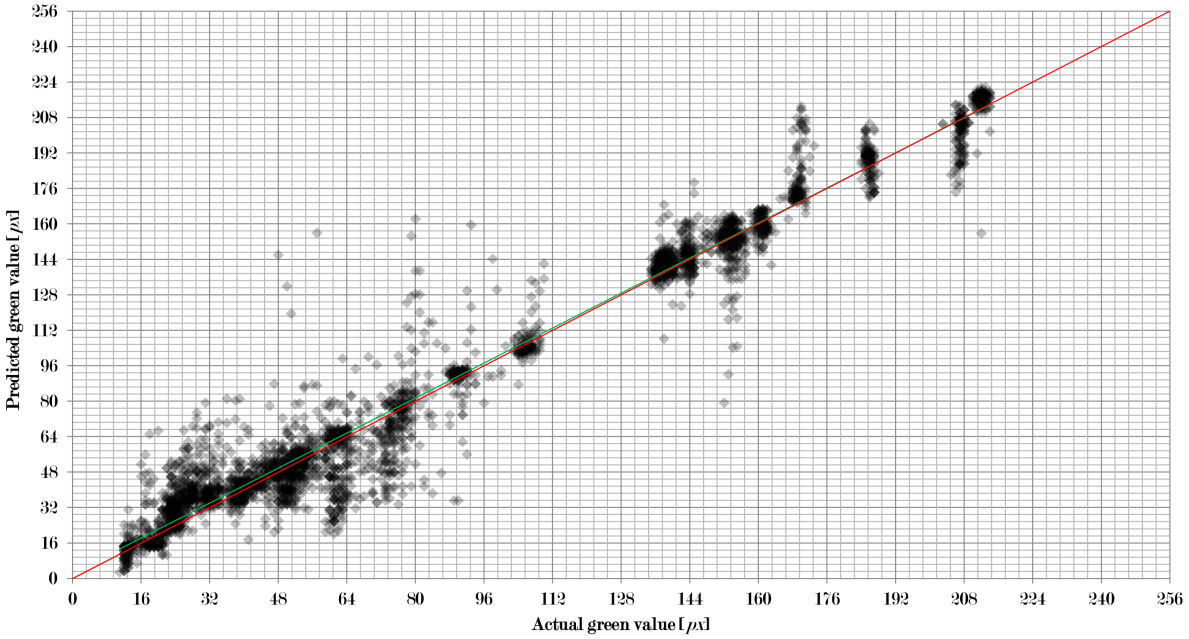


Figure 5.18: Predicted green value compared to actual green value at a water depth of 4 m and a camera distance of 5 m.

Fig. 5.18 shows how this combination of learning machines and input methods works. As for the red channel, the results look very similar to those for 1 m camera distance. Again, also the outliers are situated in the same regions in the diagram.

In case of 8 m camera distance and 4 m water depth SVM and MLR with all input methods are of interest for further study of the green channel. In contrast to the previous two conditions k-nearest neighbour cannot be used. This was also observed for the red channel.

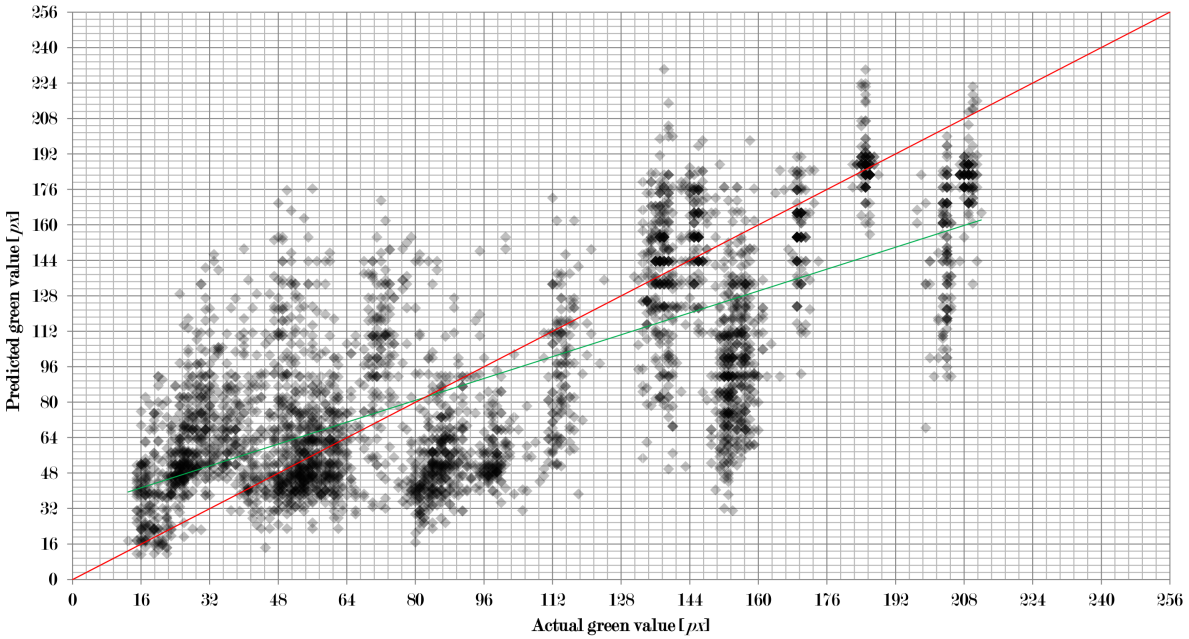


Figure 5.19: Predicted green value compared to actual green value at a water depth of 4 m and a camera distance of 8 m.

The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. MLR does not contribute. The resulting root mean square error is 39.91 px. This is slightly better than SVM with raw or SVD input which give the best single method errors.

Fig. 5.19 shows how this combination of learning machines and input methods works. As for the single methods and the red channel significant changes in the performances can be observed compared to the results for 1 m and 4 m camera distance. The points are widely spread out but much less than for the red channel. The points are also much closer to the red line when compared to the results for the red channel and a clear trend can be seen. As for the red channel for smaller pixel values the green value is also more likely to be overestimated and for higher pixel values to be underestimated. However, this characteristic is not as pronounced.

Tab. 5.12 indicates that KNN and SVM with all input methods except moving averages are of interest for further study of the green channel at 8 m depth and 1 m camera distance. The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is 20.42 px. This improves the best result obtained with a single machine (SVM with FFT) by a little bit less than 1 px.

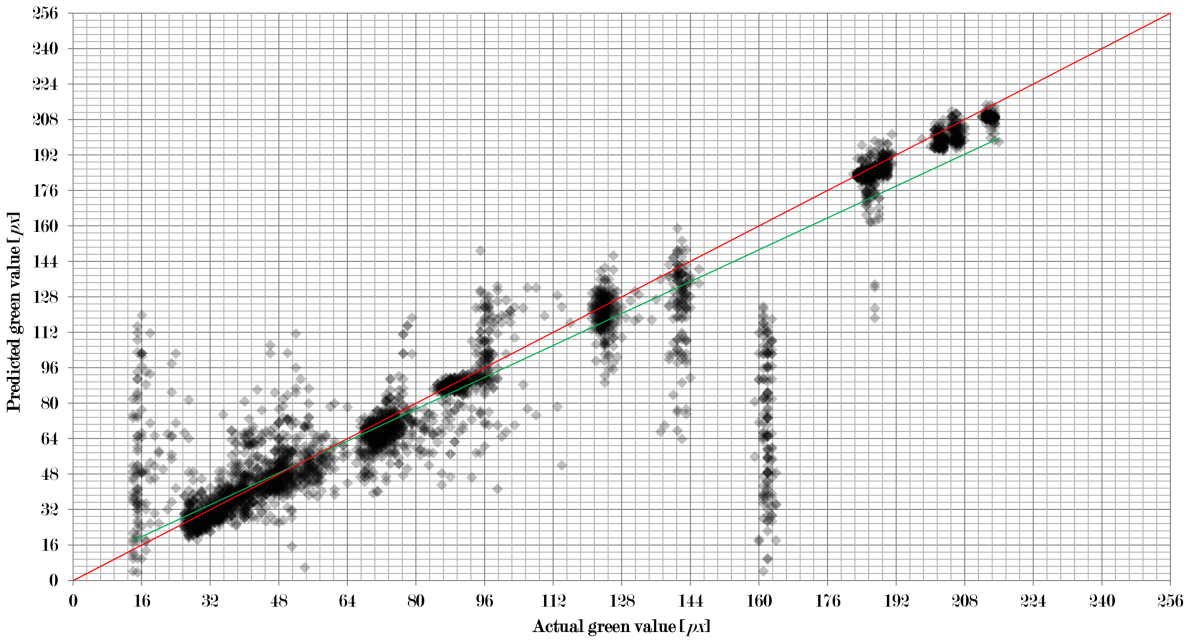


Figure 5.20: Predicted green value compared to actual green value at a water depth of 8 m and a camera distance of 1 m.

Fig. 5.20 shows how this combination of learning machines and input methods works. Again, the results are very good with some slight underestimation for values above 170 px. As for the same camera distance at 4 m water depth, the number of outliers decreases with increasing pixel value. Notable problems occur for actual green values around15 px and 162 px. For both values between 0 px and 128 px are predicted instead of the expected values. It is not clear how this occurs for these specific values.

In case of 5 m camera distance and 8 m water depth SVM and KNN with all input methods except moving averages are of interest for further study of the red channel as before. The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and

SVD input. The resulting root mean square error is 31.37 px. This improves the best result obtained with a single machine (SVM with FFT) by a little bit less than 1 px.

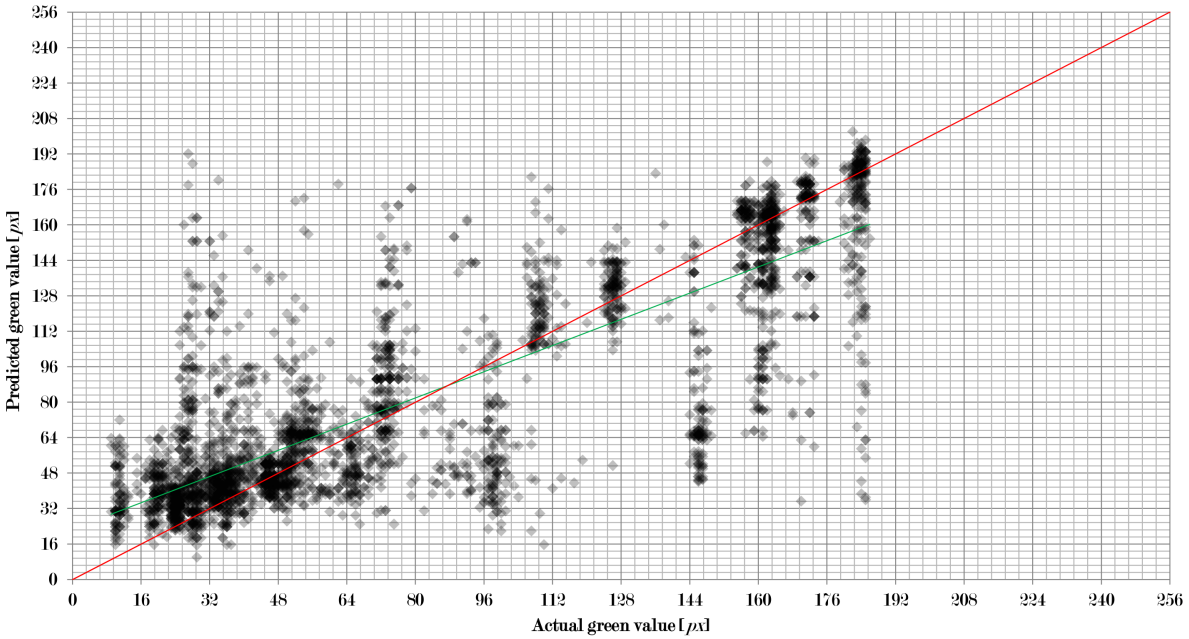


Figure 5.21: Predicted green value compared to actual green value at a water depth of 8 m and a camera distance of 5 m.

Fig. 5.21 shows how this combination of learning machines and input methods works. As for the red channel the points are more spread out than for 1 m camera distance. But the points are well-clustered near the red line and as before the green value is more likely to be overestimated for smaller pixel values and underestimated for higher pixel values.

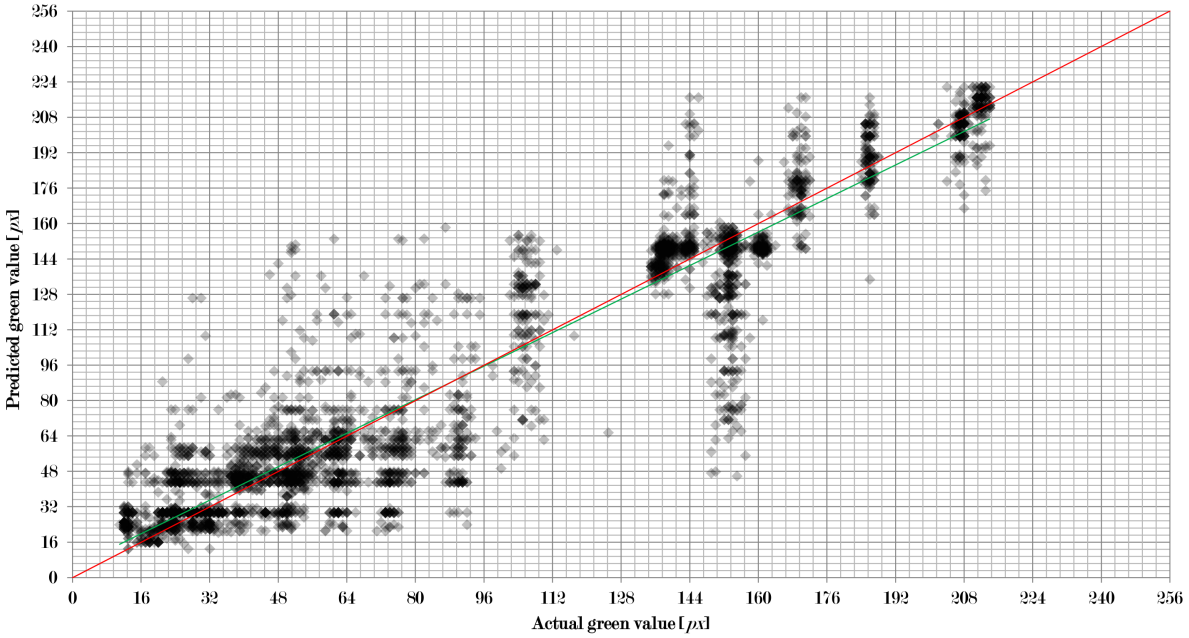


Figure 5.22: Predicted green value compared to actual green value at a water depth of 12 m and a camera distance of 1 m.

In case of 1 m camera distance and 12 m water depth KNN and SVM with all input methods

except moving averages are of interest for further study of the green channel as before. In addition multiple linear regression also shows promise in this case as the RMSE is not very high for all input methods.

The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. Multiple linear regression is not considered. The resulting root mean square error is $20.39\,px$. This improves the best result obtained with a single machine (SVM with Raw or SVD) by less than $1\,px$. It is also slightly better than the combined results for $1\,m$ camera distance at $8\,m$ water depth.

Fig. 5.22 shows how this combination of learning machines and input methods works. Compared to the results at $8\,m$ water depth at the same camera distance one can observe that the problems for actual green values around $15\,px$ and $162\,px$ do not appear any more. At the same time the points are more spread out. This combinations yields the slightly better RMSE at $12\,m$ water depth. One can also see that the overestimation for small green values and the underestimation of higher values is not so apparent any more.

5.2.4 Blue Channel

Tab.'s 5.15 to 5.20 show the root mean square errors (RMSE) for all combinations of the 5 machine learning methods and the 5 different input data for the blue channel of 24 bit RGB images.

RMSE [px]	**ANN**	**KNN**	**SVM**	**BN**	**MLR**
Raw	62.40	28.06	13.86	34.21	23.77
FFT	62.57	29.16	14.07	33.81	23.77
Conv	62.45	29.04	14.30	33.58	23.77
MA	63.41	-	41.46	37.20	23.80
SVD	62.63	28.06	13.86	34.21	23.77

Table 5.15: RMSEs for the blue channel at a water depth of $4\,m$ and a camera distance of $1\,m$.

Tab. 5.15 gives the results for a water depth of $4\,m$ and a camera distance of $1\,m$. As one can see artificial neural networks (ANN) show a large RMSE independent of the input data around $63\,px$. For Bayesian networks (BN) the RMSE is mostly around $34\,px$ and for multiple linear regression (MLR) the RSME is above $23\,px$. For k-nearest neighbour (KNN) and for support vector machines (SVM) the performance depends strongly on the input data as before. For KNN the errors are mostly between $28\,px$ and $30\,px$ except with moving averages (MA) where no meaningful results could be obtained. For SVM the RMSEs are below $15\,px$ except when combined with MA where the error is above $41\,px$.

The performances are very similar to the results for the green channel. Notable exceptions are KNN where the RMSEs are much higher for the blue channel and SVM combined with MA where the RMSE is lower than for the blue channel. The results deviate quite strongly from the laboratory results due to the optical properties of the water in the tank. Also larger errors compared to the red channel are to be expected (see fig. 1.7). This is not the case. However the camera distance and the water depth are small such that the absorption is not high.

Tab. 5.16 shows the results for a water depth of $4\,m$ and a camera distance of $5\,m$. The results are almost the same as for a camera distance of $1\,m$. This was already observed for the red and

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	61.71	29.36	14.00	34.21	23.77
FFT	61.53	30.15	14.30	33.65	23.77
Conv	61.59	30.03	14.23	33.75	23.78
MA	62.28	-	41.04	38.05	23.83
SVD	61.64	29.36	14.00	34.21	23.77

Table 5.16: RMSEs for the blue channel at a water depth of $4\,m$ and a camera distance of $5\,m$.

the green channel. However, it may be noted that support vector machines seem to work quite well with convoluted data (Conv) as the RMSE is decreased by more than $1\,px$.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	61.68	-	46.94	44.75	41.63
FFT	61.65	-	42.01	44.38	41.63
Conv	61.93	-	43.18	44.50	41.65
MA	60.67	-	44.81	43.67	41.63
SVD	61.64	-	46.94	44.75	41.63

Table 5.17: RMSEs for the blue channel at a water depth of $4\,m$ and a camera distance of $8\,m$.

Tab. 5.17 shows the results for a water depth of $4\,m$ and a camera distance of $8\,m$. The results are notably different to those for smaller camera distances. The RMSEs are much larger for SVM, BN, and MLR. The strongest drop in performance can be seen for SVM such that they produce about the same quality as BN. With ANN there is not much change which was also observed for the other colour channels. No meaningful results could be obtained for KNN.

The results are very similar to those of the green channel under the same conditions with the exception of KNN. Again, this is in contrast to the behaviour as shown in fig. 1.7 as mentioned above. However, the camera distance is sufficiently large at $8\,m$ such that the stronger absorption of blue light in coastal water should be apparent. As this is not the case one has to again conclude that there are addition local factors that change the actual absorption curve. In section 5.2.3 rust was already mentioned as a possible source.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	63.80	34.63	16.22	29.14	32.18
FFT	63.87	36.39	16.78	29.08	32.18
Conv	63.90	35.65	16.47	29.08	32.18
MA	63.32	-	40.83	38.49	32.20
SVD	63.90	34.63	16.22	29.14	32.18

Table 5.18: RMSEs for the blue channel at a water depth of $8\,m$ and a camera distance of $1\,m$.

Tab. 5.18 shows the results for a water depth of $8\,m$ and a camera distance of $1\,m$. The results show an increase of the root mean square errors when compared to the results at the same camera distance and $4\,m$ water depth for KNN, SVM, and MLR. The performance of ANN remain about the same and the errors of BN are reduced. Except for BN the development is as expected from analysis of the previous results. For ANN the RMSEs are about the same as for the green channel and better as for the red channel. For KNN the blue channel gives the worst results under this condition. In contrast, the blue channel gives the best results for SVM, BN, and MLR.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	54.41	76.47	24.45	34.29	34.32
FFT	54.40	81.06	25.45	34.29	34.32
Conv	54.47	82.93	25.28	34.29	34.32
MA	57.27	-	36.49	34.82	34.35
SVD	53.94	76.47	24.45	34.29	34.32

Table 5.19: RMSEs for the blue channel at a water depth of $8\,m$ and a camera distance of $5\,m$.

Tab. 5.19 shows the results for a water depth of $8\,m$ and a camera distance of $5\,m$. One may observe some notable differences between these results and the ones for a camera distance of $1\,m$ at the same depth. First of all, the RMSE for ANN is reduced to between $53\,px$ and $58\,px$. For KNN the error is more than doubled and now between $76\,px$ and $83\,px$. SVM, BN, and MLR also show an increase of the root means square errors. However, they are not as large as for KNN. Again, the combination of camera distance and reduction of ambient lighting results in a degradation in performance with the exception of ANN.

RMSE [px]	ANN	KNN	SVM	BN	MLR
Raw	70.05	-	38.78	50.55	39.28
FFT	70.06	-	42.44	46.96	39.28
Conv	70.05	-	38.64	46.43	39.28
MA	69.24	-	46.62	44.75	39.30
SVD	70.06	-	38.78	50.55	39.28

Table 5.20: RMSEs for the blue channel at a water depth of $12\,m$ and a camera distance of $1\,m$.

Finally, tab. 5.20 shows the results for a water depth of $12\,m$ and a camera distance of $1\,m$. One can observe an increase of the RMSEs for all learning algorithms compared to the other conditions. For KNN no meaningful results are obtained. When comparing the results to those at $8\,m$ water depth the development of the RMSE is quite sensible although a slightly smaller increase may be expected for BN.

Combined Method

As for the red channel and the green channel, it is not feasible to test all possible learning machine and input method combinations for the increase of accuracy and output stability. Tab. 5.15 indicates that SVM with all input methods except moving averages are of interest for further study of the blue channel at $4\,m$ depth and $1\,m$ camera distance.

The best solution can be obtained using support vector machines combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is $13.48\,px$. This improves the best result obtained with a single machine (SVM with Raw or SVD) by less than $1\,px$.

Fig. 5.23 shows how this combination of learning machines and input methods works. The blue value of a pixel predicted by the combined system is shown on the ordinate and plotted against the actual blue value of the pixel obtained from an image under white light.

As one can see quite a large number of results are very close to the perfect prediction (red line). There are some results that deviate. But these are only a small number and a clear trend and clustering around the red line can be seen. It can also be observed that the linear fit of the data

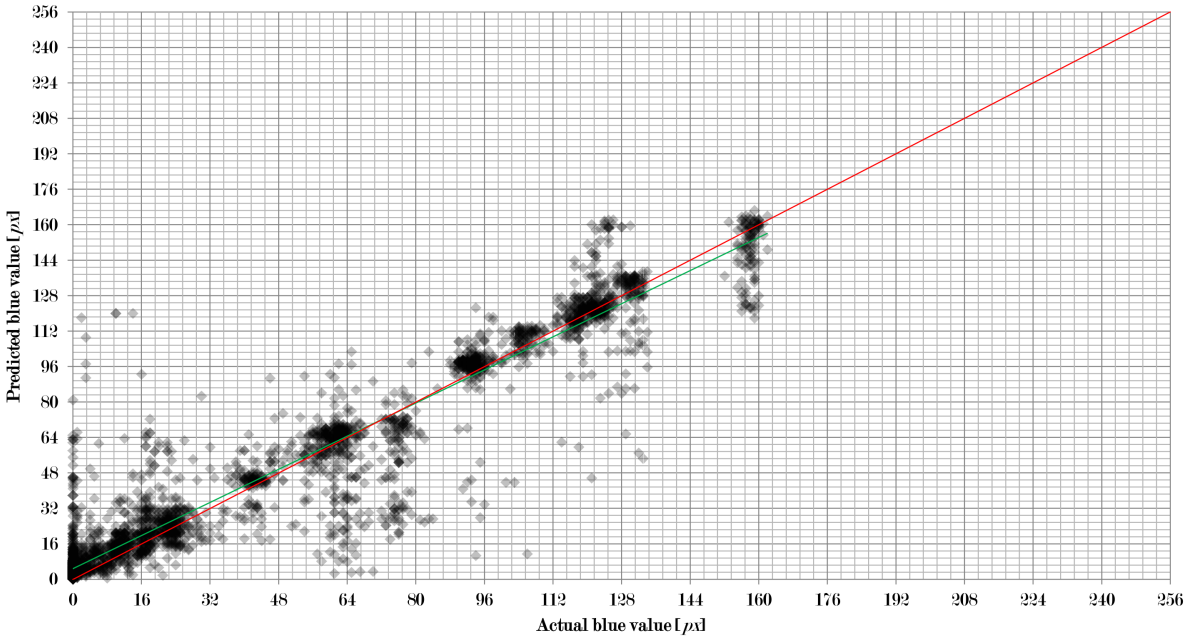


Figure 5.23: Predicted blue value compared to actual blue value at a water depth of 4 m and a camera distance of 1 m.

(green line) is almost on top of the line of perfect prediction indicates that the overall performance is excellent. Some problems can be observed for an actual blue value of 0 px where quite a few value are predicted in the range up to 64 px.

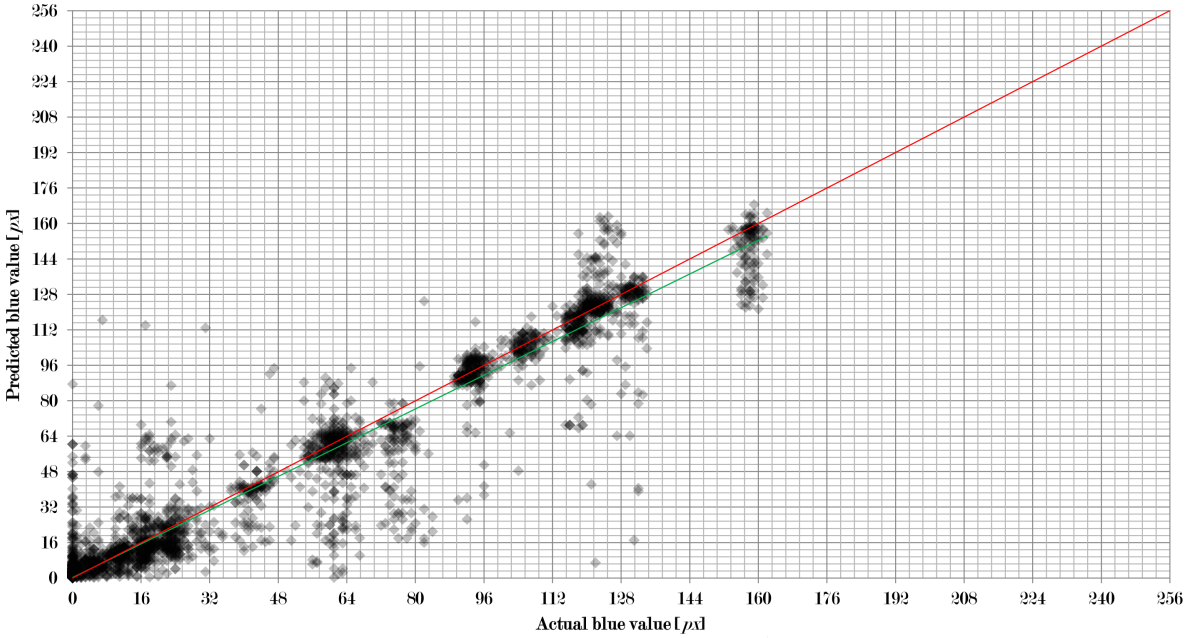


Figure 5.24: Predicted blue value compared to actual blue value at a water depth of 4 m and a camera distance of 5 m.

This was also observed for the laboratory (see fig. 3.14) where values of up to about 40 px are predicted for an actual blue value of 0 px. The overall structure is also very similar for the blue channel although the resulting root mean square error is much higher than for the laboratory. This was also observed for the other two channels.

A comparison to the results for the red channel (fig. 5.8) and the green channel (fig. 5.17) shows that there are more points which do not lie on the red line for the blue channel in both cases. There are also more outliers than for the green channel but less than for the red channel, especially for higher values.

In case of $5\,m$ camera distance at $4\,m$ water depth again SVM with all input methods except moving averages are of interest for further study of the blue channel. The best solution can be obtained using support vector machines combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is $13.50\,px$. This improves the best result obtained with a single machine by less than $1\,px$.

Fig. 5.24 shows how this combination of learning machines and input methods works. As for the red channel and the green channel, the results look very similar to those for $1\,m$ camera distance. Again, also the outliers are situated in the same regions in the diagram.

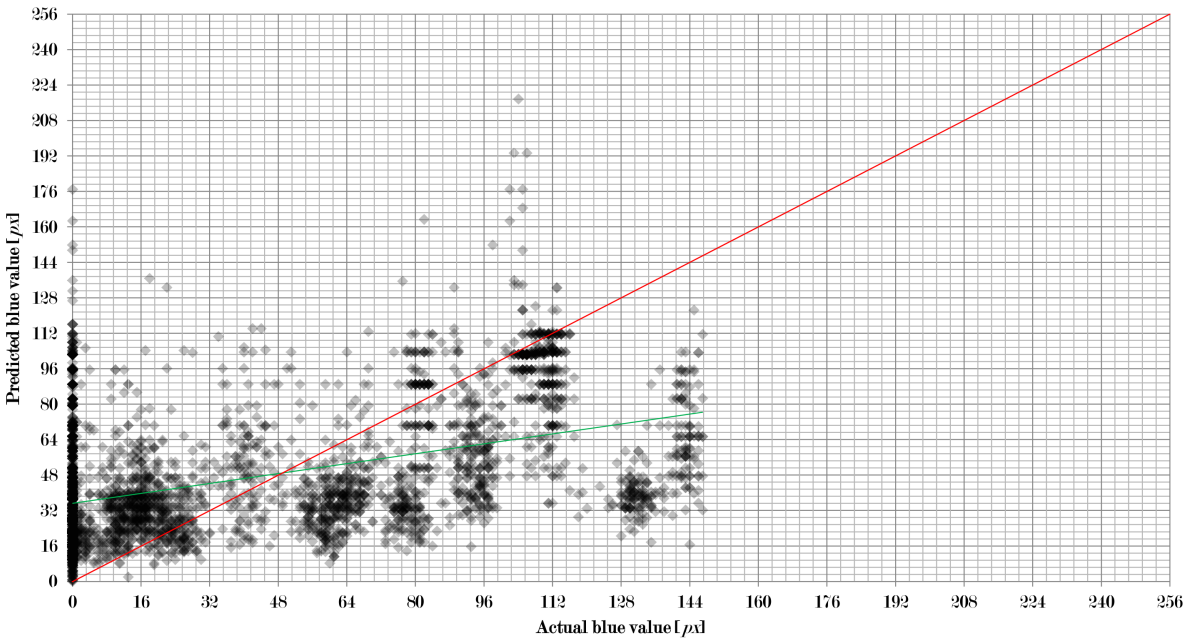


Figure 5.25: Predicted blue value compared to actual blue value at a water depth of $4\,m$ and a camera distance of $8\,m$.

In case of $8\,m$ camera distance and $4\,m$ water depth SVM, BN, and MLR with all input methods are of interest for further study of the blue channel. The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. BN and MLR do not contribute. The resulting root mean square error is $41.26\,px$. This is slightly better than MLR with raw or FFT input which give the best single method errors.

Fig. 5.25 shows how this combination of learning machines and input methods works. As for the red channel and the green channel significant changes in the performances can be observed compared to the results for $1\,m$ and $4\,m$ camera distance. The points are widely spread out but much less than for the red and the green channel. However, one can again observe the problems for an actual blue value of $0\,px$. As before quite a few points are predicted with a much higher blue value up to $120\,px$ for most points with some additional points having an even higher prediction. As before, for smaller pixel values the blue value is more likely to be overestimated and for higher pixel values to be underestimated. This characteristic is much more pronounced for the blue

channel than for the other two colour channels.

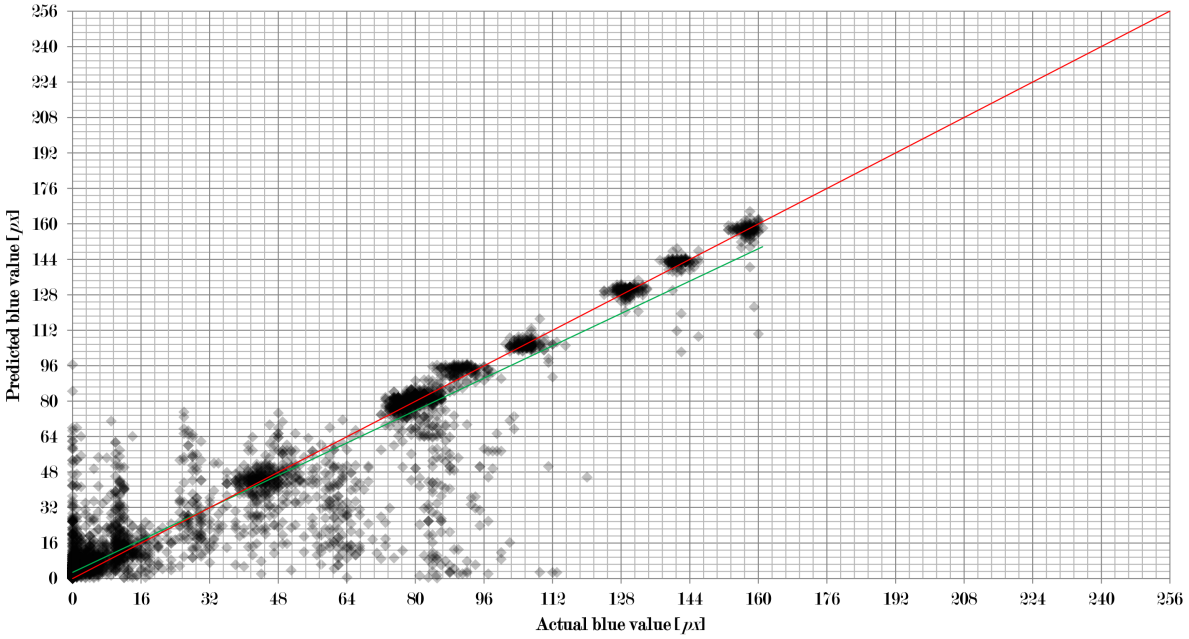


Figure 5.26: Predicted blue value compared to actual blue value at a water depth of 8 m and a camera distance of 1 m.

Tab. 5.18 indicates that SVM with all input methods except moving averages are of interest for further study of the blue channel at 8 m depth and 1 m camera distance. The best solution can be obtained using support vector machines combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is 15.63 px. This improves the best result obtained with a single machine (SVM with Raw or SVD input) by a little bit less than 1 px.

Fig. 5.26 shows how this combination of learning machines and input methods works. Again, the results are very good with some outliers for actual blue values below 100 px. Also the problem with the prediction for an actual blue value of 0 px can again be observed. Otherwise the points are well-clustered around the red line and a clear trend can be seen. Disregarding the outliers the points are even closer to the red line than for the same camera distance at 4 m water depth. However, the outliers are further out an more numerous which give a higher overall error.

In case of 5 m camera distance and 8 m water depth again SVM with all input methods except moving averages are of interest for further study of the red channel as before. The best solution can be obtained using support vector machines combined with raw, FFT, convoluted, and SVD input. The resulting root mean square error is 23.70 px. This improves the best result obtained with a single machine (SVM with Raw or SVD input) by a little bit less than 1 px.

Fig. 5.27 shows how this combination of learning machines and input methods works. As for the red channel and the green the points are more spread out than for 1 m camera distance. However, the results are much better than for the other two channels. Again the problem with the prediction for an actual blue value of 0 px can be observed. Otherwise the points are well-clustered near the red line.

In case of 1 m camera distance and 12 m water depth SVM with all input methods except moving averages are of interest for further study of the blue channel as before. In addition multiple linear regression also shows promise in this case as the RMSE is not very high for all input methods.

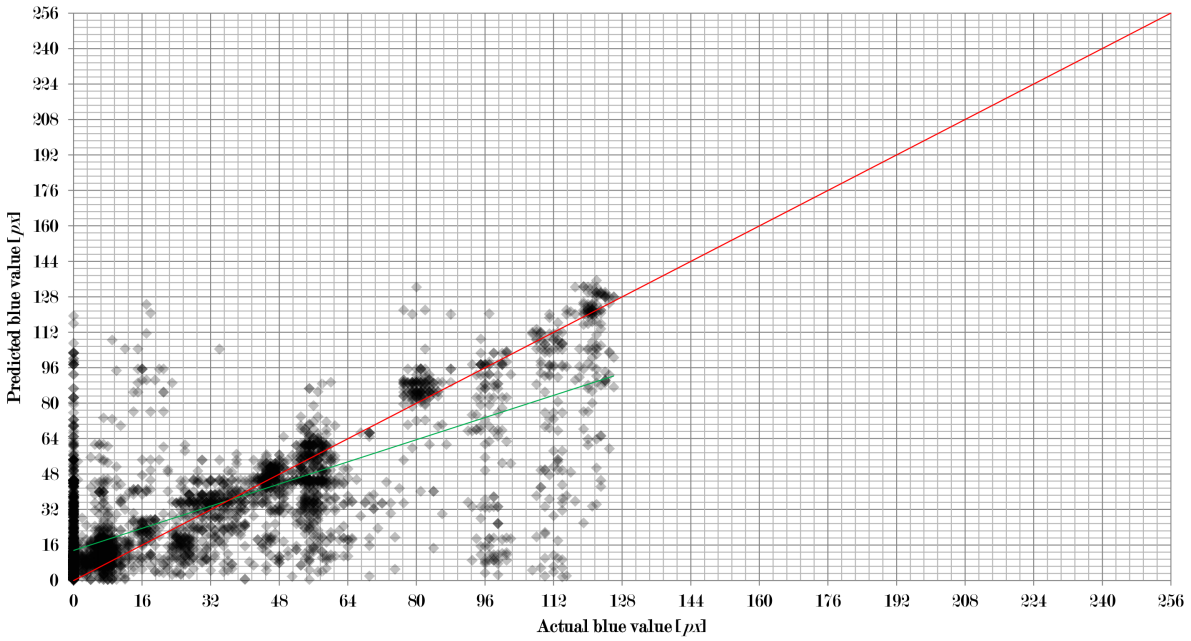


Figure 5.27: Predicted blue value compared to actual blue value at a water depth of 8 m and a camera distance of 5 m.

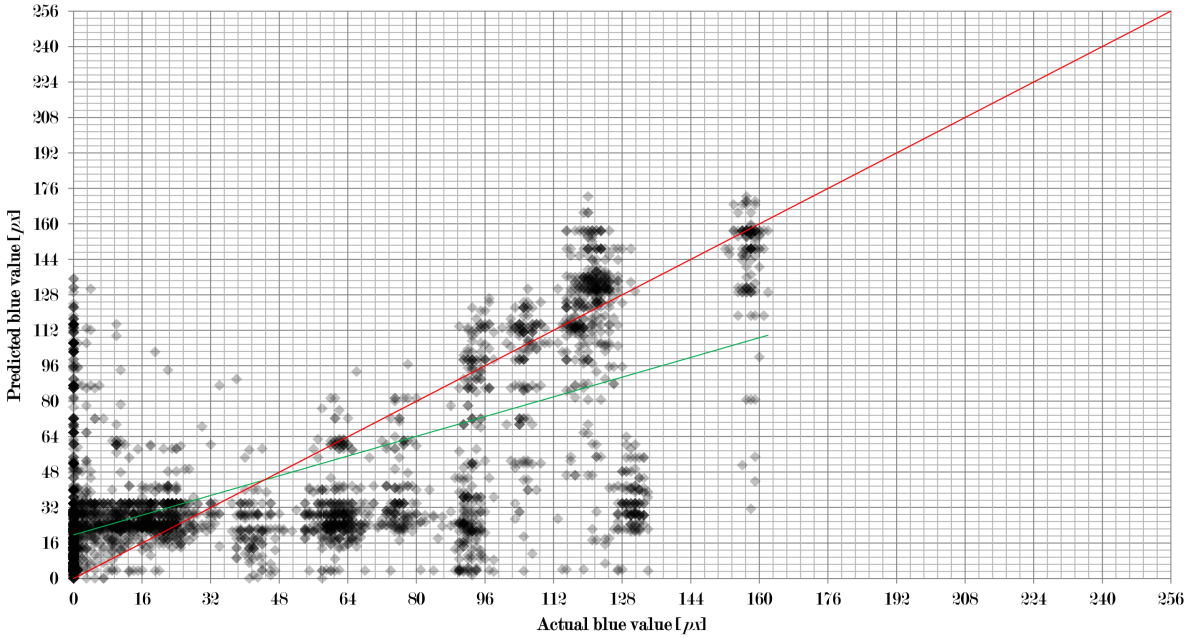


Figure 5.28: Predicted blue value compared to actual blue value at a water depth of 12 m and a camera distance of 1 m.

The best solution can be obtained using only support vector machines combined with raw, FFT, convoluted, and SVD input. Multiple linear regression is not considered. The resulting root mean square error is 37.94 px. This improves the best result obtained with a single machine (SVM with Conv) by less than 1 px.

Fig. 5.28 shows how this combination of learning machines and input methods works. The problem with the prediction for an actual blue value of 0 px can again be observed. For actual pixel values between 36 px and 84 px the prediction is quite low and the points cluster in an area around 20 px. For higher values the the prediction is much better. Also for low values the

prediction is quite good.

5.2.5 Connecting the Results of the Separate Channels

Tab. 5.21 shows which learning machines give the best results for the combined methods separately for the three colour channels. As one can see support vector machines become dominant the further away the camera is from the object and with increasing water depth. k-nearest neighbour is only used for small camera distances and small water depths. Furthermore, KNN is of real interest only for the red channel alone. For the green channel KNN is used at a camera distance of $1\,m$ and a water depth of $4\,m$ in combination with SVM and in no other case. For the blue channel only support vector machines are used. KNN does not play a role at all.

		Camera distance [m]		
		1	5	8
Water depth [m]	4	R: KNN G: KNN/SVM B: SVM	R: KNN G: SVM B: SVM	R: SVM G: SVM B: SVM
	8	R: KNN G: SVM B: SVM	R: SVM G: SVM B: SVM	
	12	R: SVM G: SVM B: SVM		

Table 5.21: Learning machines used for the combined methods.

It may therefore be concluded that support vector machines are generally the better choice for the task of obtaining the colour of objects as seen under white light from the underwater images. Only at close distances k-nearest neighbour may be used. However, this is not of interest for the task of underwater navigation and object recognition.

Having analysed the channels separately it is also necessary to discuss the colours produced by the algorithms from the underwater images. Due to the very large number of colour patches under different conditions only the worst case (water depth $12\,m$, camera distance of $1\,m$) is discussed here for selected colour patches. The observations made can also be applied to the other cases in light of the tables and figures above.

Tab. 5.22 shows the results for several colour patches. The left column shows colour patches as they appear in the laboratory under white light. In the middle column the same patches are given as they look in the underwater environment in the water tank at $12\,m$ water depth and $1\,m$ camera distance. The right column shows the reconstruction from the algorithms.

The first general observation that can be made is that there is no perceptible difference between the colour patches in the underwater environment. So there is very little information contained in the images which can be used for the reconstruction. In the first experiments in the laboratory some variations between the different colours where visible (see fig.'s 3.15 to 3.21). This is not the case for the underwater condition as presented in tab. 5.22.

The second general observation that can be made from tab. 5.22 is that bright colours appear darker in the reconstruction than under laboratory conditions. This is best see in the red colour

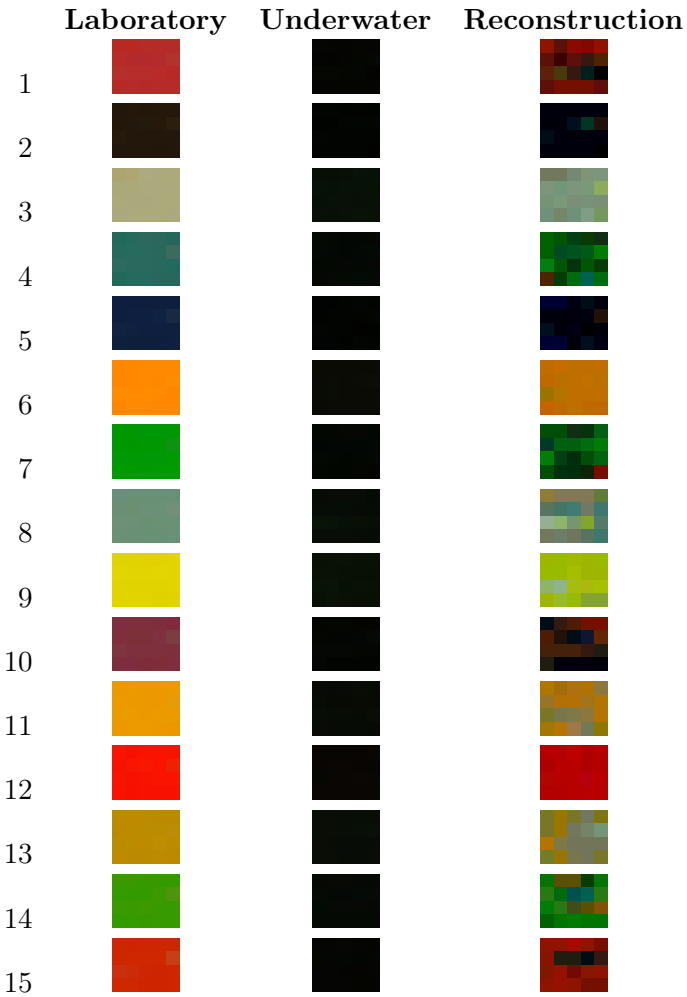


Table 5.22: Selected results at a water depth of 12 m and a camera distance of 1 m.

patches no. 12 and no. 15. It was already mentioned in the analysis of the colour channel that there is a tendency for low values to be overestimated and high values to be underestimated. So this is a direct visual consequence of this property.

Next, one can see that a high red content in the colour patches is reconstructed quite well. Patches no.'s 1, 6, 10, 11, 12, and 15 contain a significant amount of red. Especially no. 6 and no. 12 show very good results for the red content. No. 11 shows some grey instead of orange. However, this indicates problems in the combination of the channels and not in the red channel itself. In patches no. 1 and no. 10 the red content is smaller and a degradation in the quality of the reconstruction can be observed immediately.

This result is quite interesting in light of the fact that the average error in the red channel is the highest of all three colour channels. The RMSE for the red channel in the combined method is 47.37 px, compared to 20.40 px for the green channel and 37.94 px for the blue channel. However, when going back to fig. 5.13 one may observe that especially for higher red values the predicted value is very good while the mid-range is much more problematic. So this numerical result can also be seen in the visualisation.

The green content in the colour patches is also very well reconstructed. Patches no.'s 4, 7, 8, 9, and 14 contain a significant amount of green. The bright green patch no. 9 is excellent. The strong green patches no. 7 and no. 14 show more problems. However, except for some red the

resulting colour patches appear also green in the reconstruction even though the reconstruction is darker. But this was already mentioned above. In patches no. 4 and no. 8 the green content is slightly lower. The resulting colour is not great. However, the green content can clearly be seen in the reconstruction. So the problem lies not with the green content alone. For patch no. 8 it may also be observed that the mean colour in the reconstruction actually corresponds to the colour of the patch under white light.

The visual observation with regard to the green content of the colour patches are quite sensible. Green light has the longest penetration depth in coastal water. And the green colour channel shows the best overall results with a much lower RMSE than the other two colour channels. Hence, the results are to be expected.

With regard to the blue content in the colour patches one can make the following observations. High blue values are to be found in patches no.'s 4, 5, and 8. Patch no. 5 is a dark blue in the laboratory with no red and green content and a small amount of blue in the range of $64\,px$. When looking at fig. 5.28 one may observe that in this range of the actual blue value the reconstruction results in significantly lower values around $20\,px$.

Patches no. 4 and no. 8 are much brighter and therefore contain more blue. In patch no. 4 the blue value is around $93\,px$. From fig. 5.28 one can expect a strong underestimation of the blue value. This is also apparent in tab. 5.22. In patch no. 8 the blue content is even higher and around $116\,px$. In this range the reconstruction of the blue value is expected to be very good. This is the case. One may observe some grey in patch no. 8. However, the green value and the blue value are very similar in this case. As both are predicted quite well the problem is more likely to be found in the red channel. One expects a lower content of red compared to the other two colours. As there is some grey the red value is overestimated in this case.

Generally, the visual results show that the reconstruction of the colours is quite good even in the severe optical conditions under which the images are obtained. For the lower water depths the results are even better and in can be expected that for clear ocean water the errors are much smaller than the ones obtained in the experiment.

6 Conclusion

6.1 Summary

Within this thesis physics-based models for autonomous underwater vehicle navigation were developed with an emphasis on fast exploratory AUVs with cruising speeds in the range of $5\,kn$ to $20\,kn$. Furthermore, a method for the colour reconstruction of underwater images was proposed and developed.

In the first part of the thesis the applicability of different learning methods for determining flow parameters of a surrounding fluid from pressure on an AUV body were tested based on numerous computational fluid dynamical simulations and using pressure data from 56 specified points on the surface of the AUV. These points were situated on four lines along the starboard side, the port side, the top, and the bottom of the AUV body. It was shown that a combination of support vector machines was an excellent choice to perform this task.

The forward / backward component of the flow velocity could be determined with a root mean square error of $0.99\,kn$ for a velocity range of $5\,kn$ backward to $20\,kn$ forward. The best results were obtained by using the raw pressure data, pressure data with frequency noise removed through fast Fourier transform, and data smoothed through convolution with a Gaussian kernel as input for the learning machines. It could be observed that in the range of $5\,kn$ backward to $5\,kn$ forward it is very difficult to reconstruct the forward / backward component of the flow velocity from the pressure. However, a clear trend could be observed. For a forward velocity of more than $5\,kn$ the results are very good. Only for some but few unique flow situations the determination of the flow velocity was not as accurate in this range.

The sideward component of the flow velocity could be determined with a root mean square error of $0.61\,kn$ for a velocity range of $5\,kn$ port to $5\,kn$ starboard. The best results were obtained by using the raw pressure data, pressure data with frequency noise removed through fast Fourier transform, data smoothed through convolution with a Gaussian kernel, data smoothed with moving averages, and data smooth by removing small singular values through singular value decomposition as input for the learning machines. It could be observed that in the range of $2.5\,kn$ port to $2.5\,kn$ starboard it is very difficult to reconstruct the sideward component of the flow velocity from the pressure. However, a clear trend could be observed for this velocity component as well. For a velocity of more than $2.5\,kn$ to either side the results are very good. Only for some unique flow situations the determination of the flow velocity was not as accurate in this range.

The upward / downward component of the flow velocity could be determined with a root mean square error of $0.40\,kn$ for a velocity range of $5\,kn$ upward to $5\,kn$ downward. The best results were obtained by using the same input methods for the learning machines as for the sideward component. It could be observed that in the range of $1.5\,kn$ upward to $1.5\,kn$ downward it is very difficult to reconstruct the upward / downward component of the flow velocity from the pressure. But again, a clear trend could be observed. For an upward / downward velocity of more than $1.5\,kn$ the results are very good. Again, only for few unique flow situations the determination of the flow velocity was not as accurate in this range.

With the findings from the simulations the position of pressure measurement points was then

adjusted such that the most significant pressure changes due to changing flow velocities could be captured. The measurement point were then situated on 3 rings, one near the nose, one in the middle section, one near the aft, and on the top and bottom of each fin of the AUV. Each set contained 8 points. This reduced the number of measurement points to 32. It was then shown that also for the improved setup support vector machines were the best choices for the given task. However, fewer machines were required in this case. Only the raw pressure data and pressure data with frequency noise removed through fast Fourier was required as input for the learning machines.

The forward / backward component of the flow velocity could be determined with a root mean square error of $1.13\,kn$ for a velocity range of $5\,kn$ backward to $20\,kn$ forward. It could be observed that the prediction by the learning machines was much better than for the original setup. However, there were more extreme outliers resulting in a higher root means square error. The improved setup also had more difficulties with an AUV moving backward.

The sideward component of the flow velocity could be determined with a root mean square error of $0.34\,kn$ for a velocity range of $5\,kn$ port to $5\,kn$ starboard. This value was $45\,\%$ lower than for the original setup. The number of outliers was significantly reduced and the prediction by the learning machines was also much better in general.

The upward / downward component of the flow velocity could be determined with a root mean square error of $0.37\,kn$ for a velocity range of $5\,kn$ upward to $5\,kn$ downward. Also this value was lower than for the original setup. The number of outliers was significantly reduced and the prediction by the learning machines was much better in general as well.

In the second part of the thesis different learning methods were applied for the reconstruction of underwater images. First, laboratory tests were performed using a special light source with a mean wavelength of $\bar{\lambda} = 498\,nm$ imitating underwater lighting conditions. It was shown that a combination of the k-nearest neighbour method and support vector machines yields excellent results.

The red component of the colour could be determined with a root mean square error of $5.27\,px$ based on a 24 bit image. The best results were obtained by using the raw image data, image data with frequency noise removed through fast Fourier transform, and data smoothed through convolution with a Gaussian kernel as input for support vector machines. It could be observed that the reconstruction of the red component was very good with few outliers in the prediction. Also, a clear trend could be observed.

The green component of the colour could be determined with a root mean square error of $3.95\,px$ based on a 24 bit image. The best results were obtained by using the raw image data, image data with frequency noise removed through fast Fourier transform, data smoothed through convolution with a Gaussian kernel, and data smooth by removing small singular values through singular value decomposition as input for support vector machines. It could be observed that the reconstruction of the green component was even better than for the red component. The lowest errors could be observed for medium green values. However, in the middle range the number of outliers in the prediction is also highest.

The blue component of the colour could be determined with a root mean square error of $2.85\,px$

based on a 24 bit image. The best results were obtained by using the raw image data, image data with frequency noise removed through fast Fourier transform, data smoothed through convolution with a Gaussian kernel, data smoothed with moving averages, and data smooth by removing small singular values through singular value decomposition as input for support vector machines. It could be observed that the reconstruction of the blue component was the best as was expected from the lighting conditions.

The results were then combined and applied to several images of coloured objects. It was shown that reflections due to the focused light source reduced the performance of the learning machines. Also the distinction between red objects and black or dark grey objects was very difficult and sometimes led to inappropriate colours in the reconstructed image.

Based on these results an experimental verification was performed under severe conditions in murky water of a diving basin with specially prepared colour patters. It was shown that the k-nearest neighbour method gave very good results for small distances between the object and the camera and for small water depths in the red channel. For higher distances, water depths, and for the other colour channels support vector machines were the best choice for the reconstruction of the colour as seen under white light from the underwater images. It was also shown that a mixture of pre-processing methods is required to get better results. Furthermore, the reconstruction under real conditions was much more difficult than in the laboratory.

The red component of the colour could be determined with a root mean square error of $14.39\,px$ based on a 24 bit image at $4\,m$ water depth and a camera distance of $1\,m$. At this water depth the RMSE increased to $58.55\,px$ at $8\,m$ camera distance. With increasing water depth the RMSE also increased to up to $47.37\,px$ at $12\,m$ depth and $1\,m$ camera distance.

The green component of the colour could be determined with a root mean square error of $9.64\,px$ based on a 24 bit image at $4\,m$ water depth and a camera distance of $1\,m$. At this water depth the RMSE increased to $39.92\,px$ at $8\,m$ camera distance. With increasing water depth the RMSE also increased to up to $20.39\,px$ at $12\,m$ depth and $1\,m$ camera distance. So the reconstruction of the green channel was much better than the red channel.

The blue component of the colour could be determined with a root mean square error of $13.48\,px$ based on a 24 bit image at $4\,m$ water depth and a camera distance of $1\,m$. At this water depth the RMSE increased to $41.26\,px$ at $8\,m$ camera distance. With increasing water depth the RMSE also increased to up to $37.94\,px$ at $12\,m$ depth and $1\,m$ camera distance. So the reconstruction of the blue channel was worse than the green channel but better than the red channel. This was to be expected from the conditions on site as the water was similar to coastal water.

The results were then combined and the resulting colours analysed. It was shown that even under extreme conditions the reconstruction of the colours was quite good although the root mean square errors of the colour channels were high. It could be observed that bright colours appear darker in the reconstruction. It could also be seen that high colour contents were easier to reconstruct.

6.2 Outlook

With regard to the reconstruction of the flow around an AUV body the approach presented currently uses the original pressure data without any additional filtering except for the removal of frequency noise via fast Fourier transform. In the next step pre-filters should be included into the algorithm to remove unwanted influences due to outliers and measurement errors and hence make the algorithm more robust. Furthermore, the approach should be implemented into a system for testing under real conditions. It is also of interest to find a general approach to placing the pressure measurement points which can be automated such that for any given AUV the best setting can be found without having to analyse the pressure data by hand.

In addition the accuracy of the approach should be improved for small velocities such that it can also be used for slow autonomous underwater vehicles with speeds in the range of up to $3\,kn$. At these flow velocities fluctuations need to be considered. The reason is that although the fluctuations are smaller for lower velocities they become large relative to the flow velocity and therefore have a significant influence on the results. In this approach Reynolds-averaged Navier–Stokes equations (RANS) were used which are not good at capturing these fluctuations. Hence, in the next step large eddy simulations (LES) should be performed. In addition the near-wall regions need to be resolved and analysed in detail with respect to the fluctuation of the surface pressure due to the water flow and the surface roughness.

With regard to the reconstruction of underwater images the approach presented currently uses the original image without any additional filtering besides the data filters as shown. In the next step image pre-processing algorithms should be included to remove reflection effects and to smooth the brightness in the image. It may especially be of interest to include information of neighbouring pixels for the calculation of the colour to remove single pixels which are not transformed correctly.

Furthermore, the setup should be improved by using a distributed light source. The laboratory results show that a focused light presents addition obstacles for the algorithms. In water with a significant amount of suspended particles like in the diving basin used for the experimental verification focused light for illumination is also not possible as the light is directly reflected at the particles.

The procedure can be applied also for other applications, where other ranges of wavelengths are used, to obtain the image as seen under white light from data under different lighting conditions. An example would be night sensing equipment like night-vision glasses or night vision devices in cars. Finally, the experimental setup should be implemented into a real system, e.g. underwater robots and research submarines and tested further under various conditions.

7 Appendix

7.1 Details on Mesh Statistics and Mesh Quality

This section gives a more detailed analysis and discussion of the mesh statistics and mesh quality for the flow simulations done in ANSYS CFX 15.0.

7.1.1 Element Quality

Element quality Q ranges between 0 and 1, where 0 means that the element has no volume or a negative volume, and 1 means that it is a perfect tetrahedron [13] which is to be prefered. It is calculated by

$$Q = CV \left(\sum_i l_i^2 \right)^{-\frac{3}{2}} \tag{7.1}$$

where V is the volume, l_i are the edge lengths, and pre-factor C for a tetrahedron is

$$C = 124.70765802 \tag{7.2}$$

Tab. 7.1 gives some key values of the element quality for the mesh used in this thesis. As one can see there are no problematic elements and the overall quality is high.

Minimum value	0.22249
Maximum value	0.99957
Mean	0.83122
Standard deviation	0.098241

Table 7.1: Analysis of the element quality.

This can also be seen in fig. 7.1 where the number of elements for 10 quality ranges is shown.

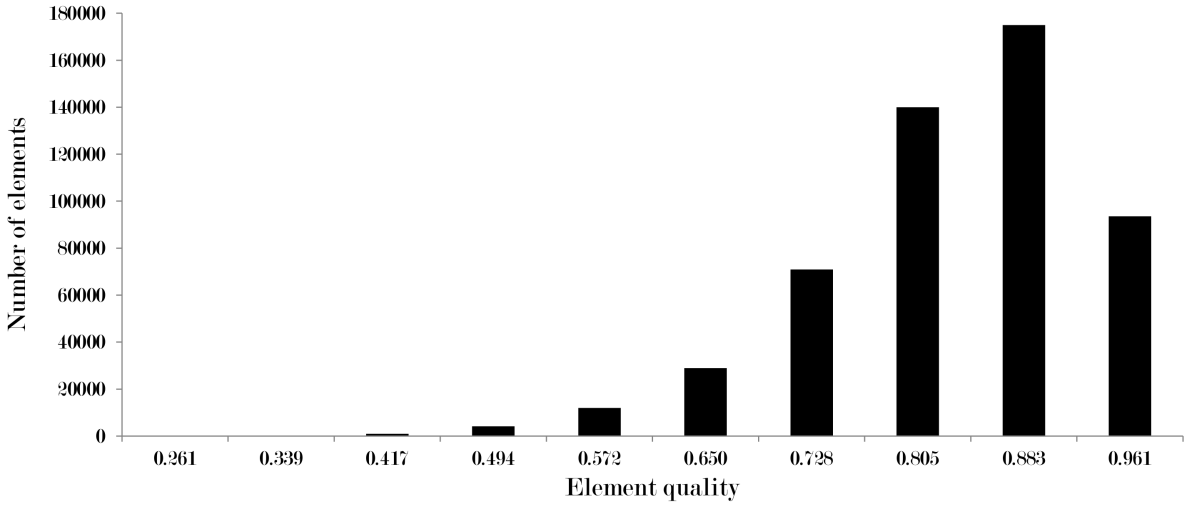


Figure 7.1: Visual representation of the element quality.

7.1.2 Aspect Ratio

The aspect ratio is a measure of how stretched an element is. A value of 1 indicates a perfect tetrahedron, while thinner, elongated tetrahedrons give higher aspect ratios [13]. A value close to 1 is desirable. Tab. 7.2 shows some key values of the aspect ratio for the mesh used.

Minimum value	1.1664
Maximum value	9.5245
Mean	1.8611
Standard deviation	0.48098

Table 7.2: Analysis of the aspect ratio.

A maximum value of 9.5245 is not perfect but as the standard deviation and fig. 7.2 show this is an exception and only very few elements have a high aspect ratio. Almost all elements have an aspect ratio below 4.5 which is very good.

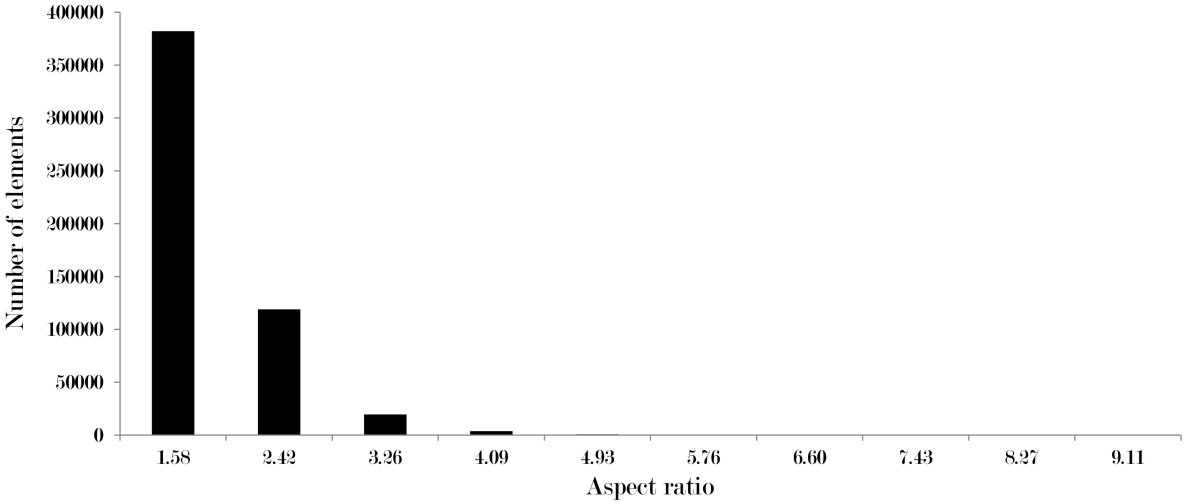


Figure 7.2: Visual representation of the aspect ratio.

7.1.3 Maximum Corner Angle

For every element the angles between adjacent edges are calculated and the maximum is determined [13]. For a tetrahedron the best possible (lowest) value is 60° for a perfect tetrahedron and the maximum value is 180° for a degenerated (flat) tetrahedron.

Minimum value	71.06°
Maximum value	159.08°
Mean	96.160°
Standard deviation	12.173°

Table 7.3: Analysis of the maximum corner angle.

Both tab. 7.3 and fig. 7.3 show that there is a wide range of angles in the mesh with most element having a maximum corner angle around 95°. The number of elements decreases strongly with increasing angle and no element element has a maximum corner angle above 160°.

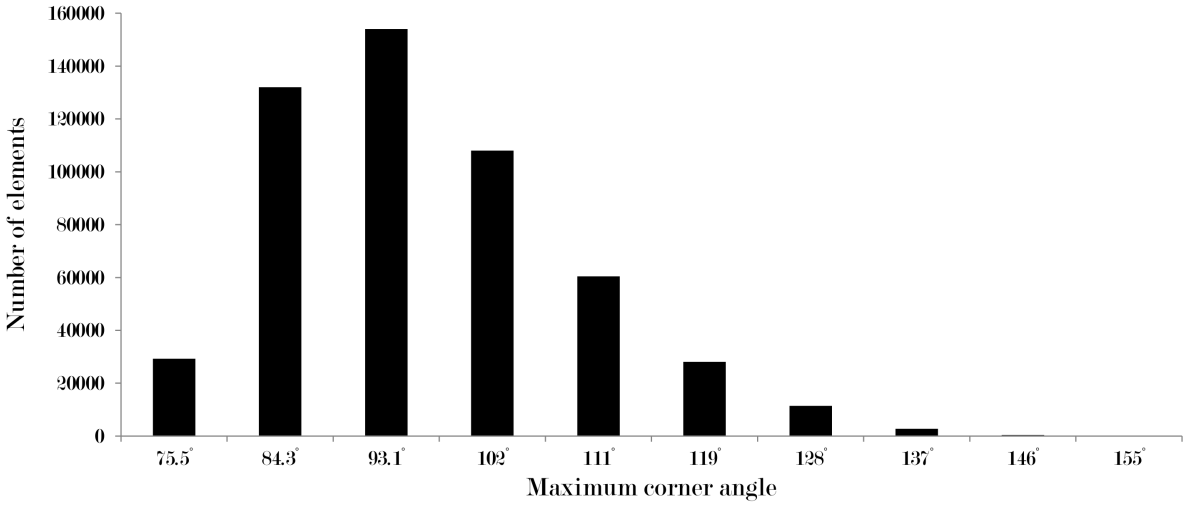


Figure 7.3: Visual representation of the maximum corner angle.

7.1.4 Skewness

Skewness S indicates how close an element's shape is to a perfect one. The measure is between 0 (equilateral) and 1 (degenerate) [13]. It is calculated as

$$S = \frac{OCS - CS}{OCS} \tag{7.3}$$

where CS is the cell size and OCS is the optimal cell size. The optimal cell size is determined as the size of an equilateral element of the same circumradius [13].

Minimum value	0.00085095
Maximum value	0.84481
Mean	0.23727
Standard deviation	0.12315

Table 7.4: Analysis of the skewness.

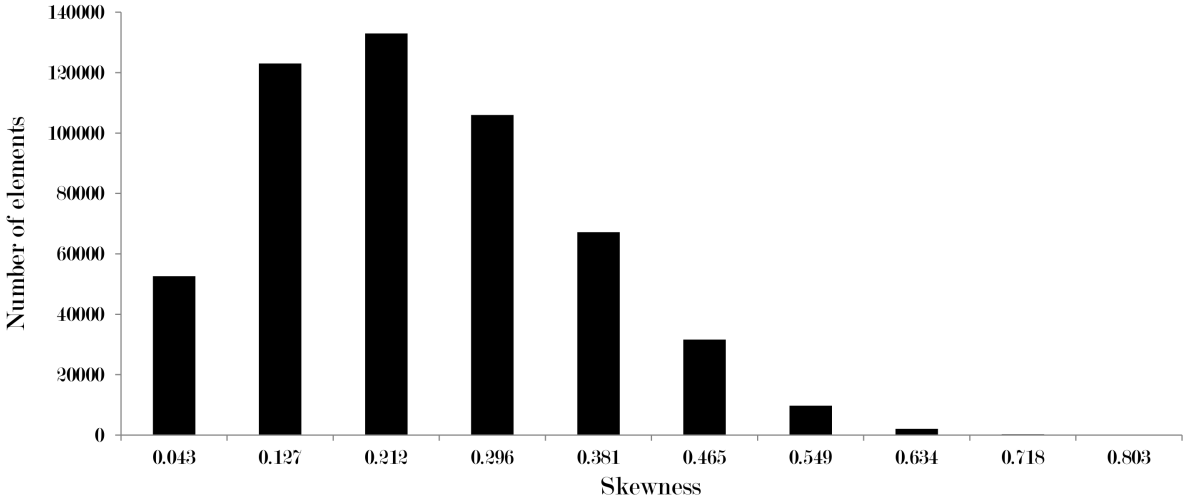


Figure 7.4: Visual representation of the skewness.

A skewness of 0.5 or better is considered good and a value of better than 0.25 said to be excellent. As tab. 7.4 and fig. 7.4 show the skewness of the elements in the mesh used is mostly very good.

Some elements show a non-optimal value. However, no element is of bad skewness (a value of 0.9 or worse) or is degenerate. So the overall skewness level of the mesh is acceptable.

7.1.5 Orthogonal Quality

The orthogonal quality is determined by the relation between the centre of an element and the centres of the surrounding elements, the relation between the centre of an element and its faces, and the face normal vectors of an element [13].

Minimum value	0.25614
Maximum value	0.99557
Mean	0.85593
Standard deviation	0.087014

Table 7.5: Analysis of the orthogonal quality.

Again the measure is between 0 and 1 with 1 being the optimal value. For the mesh used in this thesis the orthogonal quality is very good (see tab. 7.5 and fig. 7.5). No element has a quality less than 0.25 and most elements have very high quality.

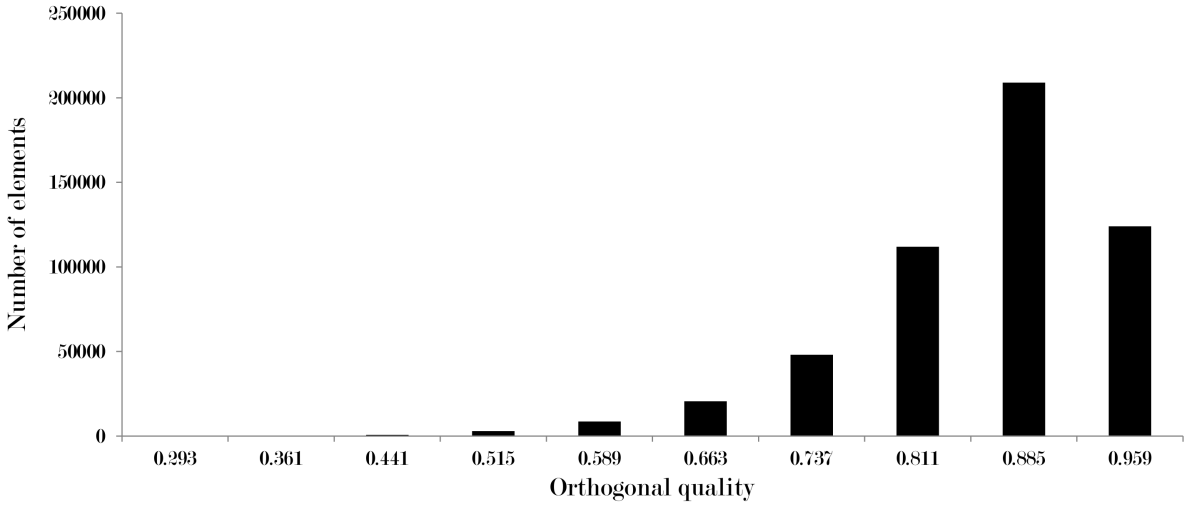


Figure 7.5: Visual representation of the orthogonal quality.

References

[1] *The American Practical Navigator.* National Imagery and Mapping Agency, United States of America, 1995.

[2] *2003 CJCS Master Positioning, Navigation, and Timing Plan.* Joint Staff, Washington, D.C., 2003.

[3] *Handbook of magnetic compass adjustment.* National Geospatial-Intelligence Agency, United States of America, 2004.

[4] *The Navy Unmanned Undersea Vehicle (UUV) Master Plan.* Department of the Navy, United States of America, November 2004.

[5] Release on the iapws formulation 2008 for the thermodynamic properties of seawater, 2008.

[6] Ajith Abraham, Bernard de Baets, Mario Koeppen, and Bertram Nickolay, editors. *Applied Soft Computing Technologies: The Challenge of Complexity*, chapter Designing Neural Networks Using Gene Expression Programming, pages 517–536. Springer-Verlag, 2006.

[7] Jonathan Adams. Alchemy or science? compromising archaeology in the deep sea. *Journal of Marine Archaeology*, 2:48–56, 2007.

[8] Charu C. Aggarwal, editor. *Data Classification: Algorithms and Applications.* CRC Press, 2014.

[9] Igor Aleksander and Helen Morton. *An Introduction to Neural Computing.* International Thomson Computer Press, 1995.

[10] Kamal M. Ali and Michael J. Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24:173–202, 1996.

[11] ANSYS, Inc. *ANSYS CFX-Solver Modelling Guide*, 2013.

[12] ANSYS, Inc. *ANSYS CFX-Solver Theory Guide*, 2013.

[13] ANSYS, Inc. *ANSYS Meshing User's Guide*, 2013.

[14] Gianluca Antonelli. *Underwater Robots.* Springer-Verlag, 2006.

[15] J. E. Bardina, P. G. Huan, and T.J. Coakley. Turbulence modeling validation, testing, and development. NASA Technical Memorandum 110446, National Aeronautics and Space Administration, 1997.

[16] Timothy J. Barth and Dennis C. Jesperson. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace Science Meeting*, 1989.

[17] Stephane Bazeille, Isabelle Quidu, Luc Jaulin, and Jean-Phillipe Malkasse. Automatic underwater image pre-processing. In *Proceedings of the Caracterisation du Milieu Marin*, Brest, France, 2006.

[18] J. G. Bellingham. *Encyclopedia of Ocean Sciences (Second Edition)*, chapter Platforms: Autonomous Underwater Vehicles, pages 473–484. Elsevier, 2009.

[19] Irad Ben-Gal. *Encyclopedia of Statistics in Quality & Reliability*, chapter Bayesian Networks. Wiley & Sons, 2007.

[20] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1996.

[21] Richard D. Blidberg, Roy M. Turner, and Steven G. Chappell. Autonomous underwater vehicles: Current activities and research opportunities. *Robotics and Autonomous Systems*, 7:139–150, 1991.

[22] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[23] E. Bovio, D. Cecchi, and F. Baralli. Autonomous underwater vehicles for scientific and naval operations. *Annual Reviews in Control*, 30:117–130, 2006.

[24] Hunter C. Brown, Ayoung Kim, and Ryan M. Eustice. An overview of autonomous underwater vehicle research and testbed at perl. *Marine Technology Society Journal*, 43(2):33–47, 2009.

[25] W. Buntine. *A Theory of Learning Classification Rules.* PhD thesis, School of Computing Science, University of Technology, Sydney, Australia, 1990.

[26] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[27] Kyle Carothers. *Light penetration in open ocean and coastal waters.* NOAA-OE, http://oceanexplorer.noaa.gov/explorations/04deepscope/background/deeplight/media/diagram3.html.

[28] V. Castelli, E.Stanley, and E. Fischer. The thermal conductivity of seawater as a function of pressure and temperature. *Deep Sea Research and Oceanographic Abstracts*, 21(4):311–319, 1974.

[29] J. A. Catipovic. Performance limitations in underwater acoustic telemetry. *IEEE Journal of Oceanic Engineering*, 15(3):205–216, 1990.

[30] Aysun Tasyapi Celebi and Sarp Ertürk. Visual enhancement of underwater images using empirical mode decomposition. *Expert Systems with Applications*, 39:800–805, 2012.

[31] Thomas S. Chance and Jay Northcutt. Deep water auv experience. Technical report, C & C Technologies, Inc., 2001.

[32] Yet-Chung Chang, Shu-Kun Hsu, and Ching-Hui Tsai. Sidescan sonar image processing: correcting brightness variation and patching gaps. *Journal of Marine Science and Technology*, 18(6):785–789, 2010.

[33] Zhe Chen, Huibin Wang, Jie Shen, Xiaofang Li, and Lizhong Xu. Region-specialized underwater image restoration in inhomogeneous optical environments. *Optik - International Journal for Light and Electron Optics*, 125(9):2090–2098, 2014.

[34] D. M. Chickering. *Learning from Data*, chapter Learning Bayesian networks is NP-complete. Springer-Verlag, 1996.

[35] Robert D. Christ and Robert L. Wernli. *The ROV Manual*. Elsevier, 2nd edition, 2014.

[36] P. Clark and R. Boswell. Rule induction with cn2: Some recent improvements. In *Proceedings of the European Working Session on Learning*, 1991.

[37] B. Claus and R. Bachmayer. Progress in the development of a navigation solution for underwater gliders. In *Proceeding of NECEC 2011*, St. John's, Newfoundland, 2011.

[38] B. Claus, L. MacNeil, and Bachmayer. Development of a magnetometry system for an underwater glider. In *Proceeding of the 17th International Symposium on Unmanned Untethered Submersible Technology 2011*, 2011.

[39] Gregory F. Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[40] Nichia Corporation. Specifications for bluish-green led, 2012.

[41] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[42] Gamini Dissanayake, Shoudong Huang, Zhan Wang, and Ravindra Ranasinghe. A review of recent developments in simultaneous localization and mapping. In *6th International Conference on Industrial and Information Systems*, 2011.

[43] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:155–161, 1997.

[44] R. Duda, J. Gaschnig, and P. Hart. *Expert systems in the micro-electronic age*, chapter Model design in the Prospector consultant system for mineral exploration. Edinburgh University Press, 1979.

[45] EIDOdata GmbH. *EIDOminer Manual*.

[46] Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. *Regression*. Springer-Verlag, 2009.

[47] Shuangshuang Fan and Craig A. Woolsey. Dynamics of underwater gliders in currents. *Ocean Engineering*, 84:249–258, 2014.

[48] Juan-Antonio Fernández-Madrigal and José Luis Blanco Claraco. *Simultaneous Localization and Mapping for Mobile Robots*. Information Science Reference, 2013.

[49] Joel H. Ferziger and Milovan Peric. *Numerische Strömungsmechanik*. Springer-Verlag, 2008.

[50] Evelyn Fix and J. L. Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[51] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[52] Nir Friedman and Moises Goldszmidt. Learning bayesian networks with local structure. In E. Horvits and F. Jensen, editors, *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 252–262, 1996.

[53] Yanrui Geng, Ricardo Martins, and Joao Sousa. Accuracy analysis of dvl/imu/magnetometer integrated navigation system using different imus in auv. In *Proceeding of the 2010 8th IEEE International Conference on Control and Automation*, pages 513–521, Xiamen, China, 2010.

[54] G. Griffiths, J. Jamieson, S. Mitchell, and K. Rutherford. Energy storage for long endurance auvs. In *Advances in Technology for Underwater Vehicles, Conference Proceedings*, London, UK, March 2004. The Institute of Marine Engineering, Science and Technology.

[55] Gwyn Griffiths. Steps towards autonomy: From current measurements to underwater vehicles. *Methods in Oceanography*, 1-2:22–48, 2012.

[56] H. Grotjans and F. R. Menter. Wall functions for general application cfd codes. In K. D. Papailiou, editor, *ECCOMAS 98 Proceedings of the Fourth European Computational Fluid Dynamics Conference*, pages 1112–1117. John Wiley & Sons, 1998.

[57] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2009.

[58] Simon Haykin. *Neural Networks and Learning Machines*. Pearson Education, 2009.

[59] David Heckermann, Dan Geiger, and David M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[60] Thomas Hill and Paul Lewicki. *Statistics: Methods and Applications*. StatSoft, Inc., 2005.

[61] J. Åhlén, E. Bengtsson, and T. Lindell. Color correction of underwater images based on estimation of diffusion attenuation coefficients. In *Proceedings of the PICS 2003: The Pics Conference, An International Technical Conference on The Science and System of Digital Photography, including the Fifth International Symposium on Multispectral Color Science*, pages 325–329, May 2003.

[62] J. Åhlén and D. Sundgren. Bottom reflectance influence on a color correction algorithm for underwater images. In *Proceedings of the 13th Scandinavian Conference on Omage Analysis (SCIA)*, volume 2749 of *Lecture Notes in Computer Science*, pages 922–926, July 2003.

[63] J. Åhlén, D. Sundgren, T. Lindell, and E. Bengtsson. Disolved organic matter impact on colour reconstruction in underwater images. In *Proceedings of the 14th Scandinavian*

Conference on Image Analysis, volume 3540 of *Lecture Notes in Computer Science*, pages 1148–1156, June 2005.

[64] Julia Åhlén. *Colour Correction of Underwater Images Using Spectral Data.* PhD thesis, Uppsala Universitet, 2005.

[65] Julia Åhlén, Ewert Bengtsson, and David Sundgren. Evaluation of underwater spectral data for colour correction applications. In *Proceedings of the 5th WSEAS Internation Conference on Circuits, Systems, Electronics, Control & Signal Processing*, Dallas, USA, November 2006.

[66] Julia Åhlén, David Sundgren, and Ewert Bengtsson. Application of underwater hyperspectral data for color correction purposes. *Pattern Recognition and Image Analysis*, 17(1):170–173, 2007.

[67] Wil S. Hylton. What happened to air france flight 447? *New York Times*, April 2011.

[68] Alexander V. Inzartsev, editor. *Underwater Vehicles.* InTech, 2009.

[69] Kashif Iqbal, Rosalina Abdul Salam, Azam Osman, and Abdullah Zawawi Talib. Underwater image enhancement using an integrated colour model. *International Journal of Computer Science*, 34(2):529–534, 2007.

[70] Khalis Isa, M. R. Arshad, and Syafizal Ishak. A hybrid-driven underwater glider model, hydrodynamics estimation, and an analysis of the motion control. *Ocean Engineering*, 81:111–129, 2014.

[71] J. D. Isdale, C. M. Spencer, and J. S. Tudhope. Physical properties of sea water solutions: viscosity. *Desalination*, 10(4):319–328, 1972.

[72] Jules S. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111, 1990.

[73] E. T. Jaynes and G. Larry Bretthorst. *Probability Theory: The Logic of Science.* Cambridge University Press, 2003.

[74] Peter Kimball and Stephen Rock. Sonar-based iceberg-relative navigation for autonomous underwater vehicles. *DeepSea Research II*, 58:1301–1310, 2011.

[75] A. D. King. Inertial navigation - forty years of evolution. *GEC Review*, 13(3):140–149, 1998.

[76] A. Korosi and B. M. Fabuss. Viscosity of liquid water from 25°c to 150°c. *Journal of Analytical Chemistry*, 40:157–162, 1968.

[77] Wai Lam and Fahiem Bacchus. Learning bayesian belief networks. an approach based on the mdl principle. *Computational Intelligence*, 10:269–293, 1994.

[78] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3(2):269–289, 1974.

[79] Donghwa Lee, Gonyop Kim, Donghoon Kim, Hyun Myung, and Hyun-Taek Choi. Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Engineering*, 48:59–68, 2012.

[80] Y. D. Eddy Lee and R. A. George. High-resolution geological auv survey results across a portion of the sigsbee escarpment. *AAPG Bulletin*, 88:747–764, 2004.

[81] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–298, 1992.

[82] Marina Magnabosco and Toby P. Breckon. Cross-spectral visual simultaneous localization and mapping (slam) with sensor handover. *Robotics and Autonomous Systems*, 61:195–208, 2013.

[83] I. Mahon. *Vision-based navigation for autonomous underwater vehicles*. PhD thesis, The University of Sydney, 2008.

[84] I. Mahon and S. Williams. Slam using natural features in an underwater environment. In *International Conference on Control, Automation, Robotics and Vision*, pages 2076–2081, 2004.

[85] Ian Mahon, Stefan B. Williams, and Matthew Johnson-Roberson. Efficient view-based slam using visual loop closures. *IEEE Transactions on Robotics*, 24(5):1002–1014, 2008.

[86] Warren McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[87] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. J. Leonard. Real-time 6-dof multi-session visual slam over large-scale environments. *Robotics and Autonomous Systems*, 61(10):1144–1158, 2013.

[88] Stephen McPhail. Autosub6000: A deep diving long range auv. *Journal of Bionic Engineering*, 6:55–62, 2009.

[89] F. J. Millero, G. Perron, and J. E. Desnoyers. Heat capacity of seawater solutions from 5° to 35°c and 0.5 to 22 chlorinity. *Journal of Geophysical Research*, 78:4499–4507, 1973.

[90] F. J. Millero and A. Poisson. International one-atmosphere equation of state of seawater. *Deep-Sea Research*, 28A(6):625–629, 1981.

[91] Frank J. Millero, Rainer Feistel, Daniel G. Wright, and Trevor J. McDougall. The composition of standard seawater and the definition of the reference-composition salinity scale. *Deep-Sea Research I*, 55:50–72, 2008.

[92] Curtis D. Mobley. *Light and Water: Radiative Transfer in Natural Waters*. Office of Naval Research, United States of America, 2nd edition, 2004.

[93] D. S. Mueller, C. R. Wagner, K. A. Oberg, and F. Rainville. *U.S. Geological Survey Techniques and Methods*, volume 3, chapter Measuring discharge with acoustic Doppler current profilers from a moving boat. U.S. Geological Survey, Reston, Virginia, 2013.

[94] Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. 6d slam–3d mapping outdoor environments. *Journal of Field Robotics*, 24(8/9):699–722, 2007.

[95] Sameer A. Nene and Shree K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. Technical report, Department of Computer Science, Columbia University, New York, 1995.

[96] G. Padmavathi, P. Subashini, M. Muthu Kumar, and Suresh Kumar Thakur. Comparison of filters used for underwater image pre-processing. *International Journal of Computer Science and Network Security*, 10(1):58–65, 2010.

[97] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. Auv navigation and localization: A review. *IEEE Jou*, 39(1):131–149, 2014.

[98] Jonathan Pearlman. Malaysia airlines mh370: Bluefin-21 submarine reprogrammed to reach record depth. *The Telegraph*, April 2014.

[99] Freddy Pøhner, Jan Ove Bakke, Kjell Eckholt Nilsen, Torbjørn Kjær, and Luciano Fonseca. Integrating imagery from hull mounted sidescan sonars with multibeam bathymetry. In *Proceedings of the U.S. Hydro 2007 Conference*, Norfolk, VA, USA, 2007.

[100] Michael E. Q. Pilson. *An Introduction to the Chemistry of the Sea.* Cambridge University Press, 2013.

[101] Tomasz Praczyk and Piotr Szymak. Decision system for a team of autonomous underwater vehicles–preliminary report. *Neurocomputing*, 74:3323–3334, 2011.

[102] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes.* Cambridge University Press, 2007.

[103] Ioseba Tena Ruiz, Sébastien de Raucourt, Yvan Petillot, and David M. Lane. Concurrent mapping and localization using sidescan sonar. *IEEE Journal of Oceanic Engineering*, 29(2):442–456, 2004.

[104] Joaquim Salvi, Yvan Petillot, and Elisabet Batlle. Visual slam for 3d large-scale seabed acquisition employing underwater vehicles. In *Interlational Conference on Intelligent Robots and Systems*, 2008.

[105] Hermann Schlichting and Klaus Gersten. *Grenzschicht-Theorie.* Springer-Verlag, 2006.

[106] Bernhard Schölkopf. *Support Vector Learning.* PhD thesis, Technische Universität Berlin, 1997.

[107] Rüdiger Schwarze. *CFD-Modellierung.* Springer-Verlag, 2013.

[108] Mae L. Seto, editor. *Marine Robot Autonomy.* Springer-Verlag, 2012.

[109] Mostafa H. Sharqawy, Hogn H. Lienhard, and Syed m. Zubair. Thermophysical properties of seawater: a review of existing correlations and data. *Desalination and Water Treatment*, 16:354–380, 2010.

[110] R. Smith, M. Salf, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 8:167–193, 1990.

[111] Raymond C. Smith and Karen S. Baker. Optical properties of the clearest natural water (200-800 nm). *Applied Optics*, 20(2):177–184, 1981.

[112] Joe Suzuki. A construction of bayesian networks from databases based on an mdl scheme. In D. Heckerman and A. Mamdani, editors, *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pages 266–273, 1993.

[113] H. U. Sverdrup, Martin W. Johnson, and Richard H. Fleming. *The Oceans Their Physics, Chemistry, and General Biology.* Prentice-Hall, New York, 1942.

[114] James L. Tangorra, Timo Gericke, and George V. Lauder. Learning from the fins of ray-finned fish for the propulsors of unmanned undersea vehicles. *Marine Technology Society Journal*, 45(4):65–73, 2011.

[115] I. Tena-Ruiz, S. Raucourt, Y. Petillot, and D. M. Lande. Concurrent mapping and localization using sidescan sonar. *IEEE Journal of Oceanic Engineering*, 29(2):442–456, 2004.

[116] Peter Varga and Raul Rojas. *Neural Networks: A Systematic Introduction.* Springer-Verlag, 1996.

[117] M. Walter, F. Hover, and J. Leonard. Slam for ship hull inspection using exactly sparse extended information filter. In *Proceedings of the IEEE International Conference on Robotic Automation*, pages 1463–1470, 2008.

[118] Jan Wendel. *Integrierte Navigationssysteme - Sensordatenfusion, GPS und Inertiale Navigation.* Oldenbourg, 2011.

[119] Paul J. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78:1550–1560, 1990.

[120] David C. Wilcox. *Turbulence Modeling for CFD.* DCW Industries, 2nd edition, 1998.

[121] Stefan B. Williams, Paul Newman, Gamini Dissanayake, and Hugh Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *International Conference on Robotics and Automation*, pages 1793–1798, 2000.

[122] Stefan B. Williams, Oscar Pizarro, Ian Mahon, and Matthew Johnson-Roberson. Simultaneous localisation and mapping and dense stereoscopic seafloor reconstruction using an auv. In *Experimental Robotics: The 11th International Symposium*, 2009.

[123] Stephen L. Wood. *Intelligent Underwater Vehicles*, chapter Autonomous Underwater Gliders, pages 505–530. I-Tech Education and Publishing, 2009.

[124] R. B. Wynn, J. Bett, A. J. Evans, G. Griffiths, V. A. I. Huvenne, A. R. Jones, M. R. Palmer, D. Dove, J. A. Howe, T. J. Boyd, and MAREMAP partners. *Investigation the feasibility of utilizing AUV and Glider technology for mapping and monitoring of the UK*

MPA network. Final report for Defra project MB0118. National Oceanography Centre, Southampton, 2012.

[125] Russell B. Wynn, Veerle A. I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brain J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous underwater vehicles (auvs): their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451–468, 2014.

[126] Ikuo Yamamoto. Research on bio-maneuvering type underwater vehicle - development of life-like swimming robotic fish. In *Proc. the 18th Ocean Engineering Symposium*, 2005.

[127] Ikuo Yamamoto. Past, present and future auv technologies. In *Masterclass in AUV Technology for Polar Science at the National Oceanography Centre, Southampton*, 2006.

[128] Dana R. Yoerger, Albert M. Bradley, Michael Jakuba, Christopher R. German, Timothy Shank, and Maurice Tivey. Autonomous and remotely operated vehicle technology for hydrothermal vent discovery, exploration, and sampling. *Oceanography*, 20:152–161, 2007.

[129] Dana R. Yoerger, Michael Jakuba, Albert M. Bradley, and Brian Bingham. Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *International Journal of Robotics*, 26:41–54, 2007.

[130] Tiedong Zhang, Wenjing Zeng, and Lei Wan. Underwater simultaneous localization and mapping based on forward-looking sonar. *Journal of Marine Science and Application*, 10:371–376, 2011.